mathematics

MDPI

Article

GWO-FNN: Fuzzy Neural Network Optimized via Grey Wolf Optimization

Paulo Vitor de Campos Souza [1,*] and Iman Sayyadzadeh [2]

[1] Intelligent Digital Agents Research Group, Fondazione Bruno Kessler, 38122 Trento, TN, Italy
[2] Rady School of Management, University of California San Diego, La Jolla, CA 92093, USA; msayyadzadeh@ucsd.edu
* Correspondence: pdecampossouza@fbk.eu

Abstract: This study introduces the GWO-FNN model, an improvement of the fuzzy neural network (FNN) architecture that aims to balance high performance with improved interpretability in artificial intelligence (AI) systems. The model leverages the Grey Wolf Optimizer (GWO) to fine-tune the consequents of fuzzy rules and uses mutual information (MI) to initialize the weights of the input layer, resulting in greater classification accuracy and model transparency. A distinctive aspect of GWO-FNN is its capacity to transform logical neurons in the hidden layer into comprehensible fuzzy rules, thereby elucidating the reasoning behind its outputs. The model's performance and interpretability were rigorously evaluated through statistical methods, interpretability benchmarks, and real-world dataset testing. These evaluations demonstrate the model's strong capability to extract and clearly express intricate patterns within the data. By combining advanced fuzzy rule mechanisms with a comprehensive interpretability framework, GWO-FNN contributes a meaningful advancement to interpretable AI approaches.

Keywords: AI; fuzzy neural networks; GWO; interpretability

MSC: 03B52

check for updates

Academic Editor: Ioannis Tsoulos

Received: 4 March 2025
Revised: 20 March 2025
Accepted: 28 March 2025
Published: 31 March 2025

Citation: de Campos Souza, P.V.; Sayyadzadeh, I. GWO-FNN: Fuzzy Neural Network Optimized via Grey Wolf Optimization. *Mathematics* **2025**, *13*, 1156. https://doi.org/10.3390/math13071156

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The rapid advancement of artificial intelligence (AI) has driven its integration across various domains, including healthcare, finance, and industrial automation. However, a significant limitation remains: the opacity of many AI models, often described as the "black box" problem, which hinders interpretability and trust in their decision-making processes [1]. With the growing emphasis on transparency and explainability in AI, there is a clear demand for models that deliver strong predictive performance while offering meaningful insights into their internal decision-making processes.

Fuzzy neural networks (FNNs) address this challenge by integrating the interpretability of fuzzy logic with the learning capabilities of neural networks. This hybrid approach allows FNNs to process uncertainty and imprecise information while maintaining structured rule-based reasoning [2]. Their architecture typically consists of fuzzification layers, logical neuron layers, and optimization mechanisms, making them suitable for tasks requiring both explainability and adaptability. FNNs combine the strengths of fuzzy logic and neural network learning, offering a notable benefit in classification scenarios by producing clear fuzzy rules that support transparent decision making. This feature is especially important in fields where understanding model reasoning is essential, such as in medical diagnostics and financial risk evaluation.

Over the past two decades, various neuro-fuzzy models have been proposed, each with distinct approaches to rule extraction and parameter optimization [3]. Comprehensive reviews [2,4] have analyzed numerous FNN methodologies, detailing their characteristics and trade-offs in terms of model complexity and computational efficiency.

FNNs possess three key properties [5]:

- Adaptive learning: FNNs continuously update their parameters in response to new data, enabling them to maintain accuracy in dynamic environments.
- Rule-based interpretability: Through fuzzy logic, FNNs generate human-readable rules, enhancing transparency and facilitating expert-driven refinements.
- Hybrid reasoning: By combining neural network learning with fuzzy inference, FNNs balance computational efficiency with qualitative decision making.

A critical component of FNNs is the management of fuzzy rules, ensuring their relevance and effectiveness in decision making. Several studies have proposed methods to assess rule significance, including evaluating activation levels [6,7], statistical impact [8], and second-order derivatives [9]. Other approaches focus on refining rule bases by pruning redundant rules or merging similar neurons [10,11], thus improving model interpretability without sacrificing predictive performance.

While many recent FNN models emphasize accuracy, particularly for streaming data applications [12], interpretability remains a challenge. Efforts such as those by Lughofer [13] seek to enhance model transparency, yet further refinements are necessary to balance interpretability and accuracy effectively.

Moreover, traditional FNN architectures predominantly utilize t-norms or multidimensional kernels, resulting in rule antecedents that are primarily "AND" operations. This constraint limits the flexibility of FNNs in representing complex relationships. Incorporating "OR" connections can significantly improve interpretability and class differentiation, a direction explored in studies on logic-based neuro-fuzzy models [14].

In summary, FNNs offer a promising solution to the challenge of explainable AI by enhancing transparency while maintaining neural network learning capabilities. Further research is needed to refine rule extraction methods and parameter tuning and develop architectures that improve both accuracy and interpretability.

1.1. Our Approach

This paper introduces an enhanced fuzzy neural network (FNN) model, GWO-FNN, building upon the foundation established by de Campos Souza [15]. The model integrates Grey Wolf Optimizer (GWO) to optimize fuzzy rule consequents and leverages mutual information (MI) to refine the initialization of input layer weights. These enhancements improve both interpretability and classification accuracy.

The GWO-FNN framework integrates the GWO, a metaheuristic optimization algorithm inspired by the hunting behavior of gray wolves, to improve the global search process and optimize fuzzy rule consequents. By integrating fuzzy logic with nature-inspired optimization, the model balances interpretability and predictive performance, resulting in more transparent classification.

1.2. Model Architecture and Interpretability Enhancements

The proposed FNN model builds on de Campos Souza's framework [15], introducing enhancements to improve interpretability and classification accuracy.

The model consists of three distinct layers, each of which serves a fundamental role in processing and decision making:

- Input layer (fuzzification process): This layer transforms numerical inputs into fuzzy values using membership functions (MFs), assigning each input a degree of member-

ship in predefined fuzzy sets. The selection and parametrization of MFs are crucial for preserving the data structure and influencing model performance.

- Hidden layer (logical neurons and rule processing): The hidden layer comprises logical fuzzy neurons [16] that apply logical operations (e.g., AND, OR) to fuzzified inputs, leveraging fuzzy rules extracted from data or expert knowledge. Each neuron represents a distinct fuzzy rule, facilitating interpretable decision making. Optimizing these neurons ensures effective rule representation and enhances system robustness.
- Output layer (defuzzification and prediction): The final layer aggregates fuzzy outputs into crisp predictions or class labels. The Grey Wolf Optimizer (GWO) is employed to optimize rule consequent weights, improving class separability and prediction accuracy. Additionally, mutual information (MI) is incorporated in the input layer to assign higher initial weights to features with greater discriminative power, enhancing interpretability.

Integrating the GWO for rule-consequent optimization refines fuzzy rule outputs, enhancing classification performance and decision reliability. Concurrently, MI-based input weight initialization ensures a more transparent decision-making process by emphasizing the most informative features. These modifications collectively enable a more adaptive and interpretable fuzzy neural network, particularly suited for complex classification tasks.

To evaluate model effectiveness, key performance metrics are analyzed, including accuracy, interpretability, and fuzzy rule alignment with data patterns and domain knowledge. This evaluation framework ensures that the model achieves not only high predictive accuracy but also delivers meaningful insights through its rule-based reasoning mechanism.

The contributions of this model can be summarized as follows:

- Enhanced rule interpretability: The model incorporates advanced metrics, such as similarity, distinguishability, and rule activation levels, ensuring that each fuzzy rule uniquely contributes to decision making. Visualization of rule activations provides additional interpretability for users.
- Mutual information for input weight initialization: MI is used to assign higher initial weights to input features with greater discriminative power, aligning fuzzy rules with meaningful input dimensions and strengthening transparency.
- Grey Wolf Optimizer for rule-consequent optimization: The GWO is applied to optimize the consequent parameters of fuzzy rules, improving classification accuracy by refining rule outputs dynamically while maintaining interpretability.
- Adaptive learning through the GWO: GWO-based optimization enables dynamic tuning of rule consequent weights in the output layer, ensuring that the weights reflect the significance of each fuzzy rule, thereby enhancing system reliability and accuracy.

Fuzzy rules within this model are assigned significance levels to ensure interpretability. Rule evaluation considers not only rule weights but also similarity and distinguishability measures, guaranteeing that each rule is distinct and meaningful. Less relevant or redundant rules can be pruned, enhancing efficiency without sacrificing performance. Conversely, critical rules are emphasized to provide a richer understanding of underlying data relationships.

Furthermore, graphical representations of fuzzy rule activations are incorporated, offering an intuitive way to understand how different inputs influence the model's predictions. This visualization bridges the gap between abstract mathematical constructs and practical interpretability.

The model is extensively evaluated on well-known binary classification datasets from the UCI machine learning repository, focusing on predictive accuracy and interpretability. By leveraging the GWO for training optimization, the model achieves an optimal balance

between accuracy and explainability. The interpretability of the model is rooted in its ability to generate and analyze fuzzy rules, providing valuable insights into data patterns and decision-making mechanisms, ensuring practical applicability for domain experts and practitioners.

Our results section details the performance of our model against existing state-of-the-art methodologies, highlighting its superior accuracy and the effectiveness of its interpretability mechanisms. By analyzing the extraction and evolution of fuzzy rules during training, we demonstrate the model's adaptability and its ability to provide a comprehensive understanding of dataset characteristics.

The knowledge extraction process is further illustrated using a sepsis identification study (originating from [17]), including the rules generated and their associated consequents. These examples illustrate not only the precision of our model's predictions but also its capacity to offer actionable insights, making it an invaluable tool for tasks requiring a high degree of accuracy coupled with deep interpretability. Through this meticulous evaluation, our model demonstrates a significant advancement in the field of FNN, offering a new paradigm for machine learning models that prioritize both performance and understanding.

In Section 2, the foundational theoretical underpinnings of fuzzy systems and interpretability are expounded upon, with a particular focus on pertinent studies concerning stress identification.

Section 3 explores the fuzzy neural network architecture, detailing the fuzzy neuron and the training procedures applied to the components and parameters of our neuro-fuzzy system. For clarity, all variables and parameters utilized in the model are systematically summarized in the nomenclature table presented in Appendix A, Table A1.

The multifaceted nature of advanced model interpretability is elucidated in Section 4.

Transitioning to Section 6, a comprehensive presentation of results is provided, including a comparative analysis with related state-of-the-art works and an exposition of the derived rules for interpretation.

Lastly, Section 8 encapsulates the principal conclusions drawn from this study.

2. Literature Review

The convergence of fuzzy logic and neural network methodologies has given rise to fuzzy neural networks (FNNs), which have emerged as powerful tools for handling complex, uncertain, and imprecise information. This hybrid approach harnesses the complementary advantages of both techniques—fuzzy systems' interpretability and neural networks' learning capabilities. In this section, we explore recent advancements in the field of FNNs, highlighting their foundational concepts, key innovations, and diverse applications within artificial intelligence.

2.1. Fuzzy Systems and Fuzzy Logic Neurons

Fuzzy systems, rooted in fuzzy logic, offer a mathematical framework for handling uncertainty by employing linguistic variables and fuzzy sets. Central to fuzzy logic are fuzzy sets, which generalize classical set theory to accommodate gradual membership degrees. Fuzzy logic neurons, inspired by biological neurons, encapsulate the principles of fuzzy logic within computational units. These neurons employ fuzzy inference mechanisms to process inputs and generate output activations, effectively capturing the nuances of uncertain data [18].

Fuzzy Sets and Developed Logic

Fuzzy sets extend classical set theory by allowing elements to have a degree of membership rather than a strict binary classification. In this work, we employ Gaussian membership functions, which provide a smooth and continuous transition between full membership and non-membership.

The membership function $\mu_A(x)$ of a fuzzy set A is defined as

$$\mu_A(x) \in [0, 1], \tag{1}$$

where x represents an element, and $\mu_A(x)$ denotes its degree of membership in set A. In classical set theory, membership is binary:

$$\mu_A(x) = \begin{cases} 1 & \text{if } x \text{ fully satisfies } A \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

However, fuzzy logic generalizes this concept, allowing membership functions to take continuous values. In this study, we use Gaussian membership functions, which are defined as

$$\mu_A(x) = \exp\left(-\frac{(x-c)^2}{2\sigma^2}\right), \tag{3}$$

where c is the center of the function, and σ controls the spread. This formulation ensures smooth transitions and avoids abrupt changes in membership values.

2.2. Fuzzy Logic Operators

In fuzzy logic systems, t-norms and t-conorms serve as foundational operators that define the core mechanisms for handling conjunction and disjunction of fuzzy truth values. T-norms correspond to fuzzy AND operations, enabling the intersection of fuzzy sets by emphasizing overlapping areas of high membership. T-conorms, in contrast, model fuzzy OR operations, supporting inclusive decision making by capturing scenarios where satisfying at least one condition is sufficient [19].

The utility of t-norms lies in their capacity to model strict conditions, aggregating evidence where multiple inputs must be jointly considered. T-conorms, on the other hand, are suited for more permissive logic, where individual favorable inputs can lead to an overall positive evaluation, offering greater adaptability in fuzzy inference mechanisms [19].

Beyond binary operations, both t-norms and t-conorms can be extended to handle multiple inputs and continuous domains, reinforcing their importance in the design of sophisticated fuzzy rule-based systems. Understanding these operators is essential for developing expressive fuzzy models capable of managing uncertainty and vagueness in real-world scenarios [20].

Among various formulations, the product (for t-norms) and the probabilistic sum (for t-conorms) are commonly adopted due to their simplicity and effectiveness [20]. These are the formulations employed in this study, following the representation outlined by [19].

$$t(x, y) = x \times y \tag{4}$$

$$s(x, y) = x + y - x \times y \tag{5}$$

Uninorms [21] extend the traditional definitions of t-norms and s-norms by introducing a more flexible structure. Unlike classical norms, which operate under fixed boundary conditions, uninorms permit the identity element g to assume any value within the unit in-

terval $[0, 1]$. This relaxation enables a generalized approach to aggregation, accommodating a broader range of logical and decision-making behaviors.

By introducing a flexible identity element g, uninorms generalize both t-norms and s-norms by allowing g to assume any value within the interval $[0, 1]$. This formulation enables a smooth transition between the behavior of an s-norm, when $g = 0$, and a t-norm, when $g = 1$. It is important to highlight that binary operators (BOs), defined as mappings from $[0, 1]^2$ to $[0, 1]$, retain essential properties such as monotonicity, commutativity, and associativity. The specific uninorm U adopted in this study is defined as follows [22]:

$$U(x, y, g) = \begin{cases} g\, T(\frac{x}{g}, \frac{y}{g}), if\ y\ \in [0, g] \\ g + (1 - g)\, S\,(\frac{x-g}{1-g}, \frac{y-g}{1-g}),\ if\ y\ \in\ (g, 1] \end{cases} \tag{6}$$

In the expressions, T and S denote generic t-norm and t-conorm operators, respectively, both of which satisfy the properties of commutativity and associativity [23,24]. A key feature of the uninorm structure is that the emphasis on logical conjunction (AND) or disjunction (OR) depends directly on the value assigned to the identity element g. For example, as g approaches 0, the first condition in Equation (6) is rarely fulfilled, causing the uninorm U to behave primarily like an OR connection. This characteristic provides substantial flexibility, enabling the incorporation of varying degrees of AND/OR emphasis within a single rule antecedent, and allowing different connection types to coexist and adapt across components of the same fuzzy rule.

The 3D visualizations presented in Figure 1 illustrate the behavior of uninorm, t-norm, and s-norm (t-conorm) operators, utilizing Equations (4)–(6) to model fundamental logical operations in fuzzy logic. The t-norm reflects the intersection or logical "AND", where darker regions in the graph indicate a lower degree of association, while lighter areas denote high concordance between input values. The s-norm models the union or logical "OR", with color variations representing the inclusiveness of values. The colors in the graphs serve as an intuitive visual representation of the aggregation performed by each operator, highlighting the smooth transition between low and high association as input values vary.

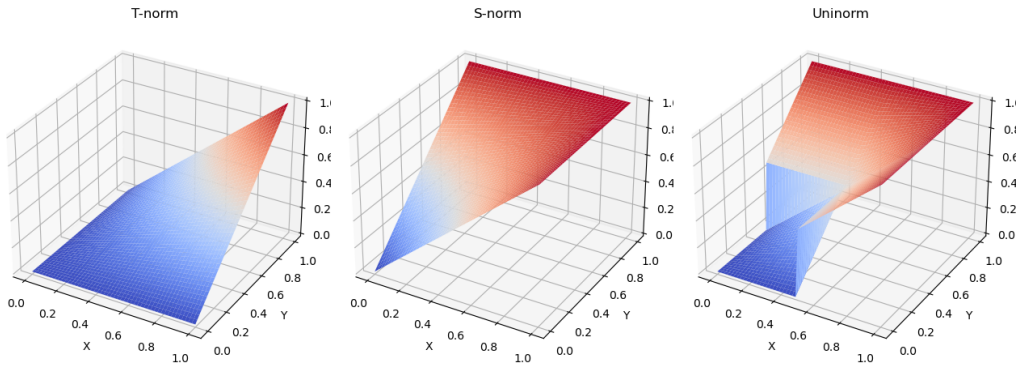


Figure 1. Three-dimensional visualization of aggregation operations: t-norm, t-conorm, and uninorm.

2.3. Fuzzy Logic Neurons

Fuzzy logic neurons extend traditional neural network architectures by incorporating fuzzy inference mechanisms into their computational model. These neurons employ fuzzy rules to map input patterns to output activations, enabling nonlinear mapping in high-dimensional spaces. By leveraging fuzzy logic principles, fuzzy logic neurons excel in domains characterized by uncertainty and incomplete information, offering enhanced flexibility and adaptability [18].

The output activation y of a fuzzy logic neuron can be computed using fuzzy inference [18]:

$$y = \sum_{j=1}^{m} w_j \cdot \mu_{B_j}(x) \tag{7}$$

where w_j represents the weight associated with fuzzy rule B_j and $\mu_{B_j}(x)$ denotes the degree of activation of rule B_j given input x.

Fuzzy logic neurons leverage binary operators to aggregate fuzzy inputs and compute output activations. By incorporating binary operators within their computational model, these neurons enable the synthesis of complex fuzzy rules and the integration of heterogeneous sources of information. This flexibility in rule synthesis enhances the adaptability of FNNs, enabling them to effectively model nonlinear relationships in diverse datasets.

2.3.1. AndNeuron and OrNeuron

Logical neurons function as computational units that integrate logical processing with learning capabilities through a system of fuzzy rules. They can be conceptualized as multivariable nonlinear transformations mapping from [0, 1] to $[0,1]^n$ [18]. Consequently, neurons for logical and and or operations aggregate fuzzy relevance values $\mathbf{a} = [a_1, a_2, \ldots, a_3, \ldots a_N]$ by initially combining them individually with corresponding weights $\mathbf{w} = [w_1, w_2, \ldots, w_3, \ldots w_N]$, where $\mathbf{a}$, $\mathbf{w} \in [0,1]^n$. The amalgamation of these results is carried out according to the methodology outlined in [18]:

$$\mathbf{z} = AND(\mathbf{w;a}) = T_{i=1}^{n}(w_i \; s \; a_i) \tag{8}$$

$$\mathbf{z} = OR(\mathbf{w;a}) = S_{i=1}^{n}(w_i \; t \; a_i) \tag{9}$$

where S and s are s-norm (typically product is used) and T and t are t-norm (typically probabilistic sum is used).

2.3.2. UniNeuron

The UniNeuron leverages uninorm concepts [21] to perform simplified operations based on fuzzy neuron activation functions, allowing for the utilization of both and and or neuron concepts [22]. The UniNeuron's processing unfolds across two levels, denoted as L_1 and L_2. In L_1, individual calculations of input signals are conducted using weights, while in L_2, a global aggregation of results from all L_1 components takes place. T-norms and s-norms are employed in these neurons to compute outputs. The essential steps involve converting each (a_i, w_i) pair into a single value b_i = **h** (a_i, w_i), followed by aggregating all these values using **U** ($b_1, b_2 \ldots b_n$), where n is the number of inputs. The transformation of inputs and weights into individual values is achieved through the application of the p function, as described by [25].

$$p(w, a, g) = wa + \bar{w}g \tag{10}$$

Using the weighted aggregation reported above, the UniNeuron can be written as

$$z = UNI(w; a; g) = U_{i=1}^{n} p(w_i, a_i, g) \tag{11}$$

In Figure 2, we present a three-dimensional visualization of the fuzzy logic operations conducted by AndNeuron, OrNeuron, and UniNeuron, which employ t-norm, s-norm, and Uninorm operations, respectively, as their fundamental aggregation mechanisms. The AndNeuron is visualized through the operation Equation (8), representing the logical "AND" by the product of fuzzy outputs and weights. The OrNeuron, depicted as (Equation (9)), models the logical "OR" through a weighted sum, normalized by the sum

of weights to maintain balance. Lastly, the UniNeuron showcases a dynamic aggregation strategy (Equation (11)), which adaptively switches between "AND" and "OR" logic based on the input values and a parameter g, thus offering a nuanced approach to fuzzy logic integration within neural structures. The visualization emphasizes the distinct aggregation patterns and operational dynamics, with color gradients indicating the degree of association or aggregation achieved by each neuron type under varying input conditions and weight configurations.

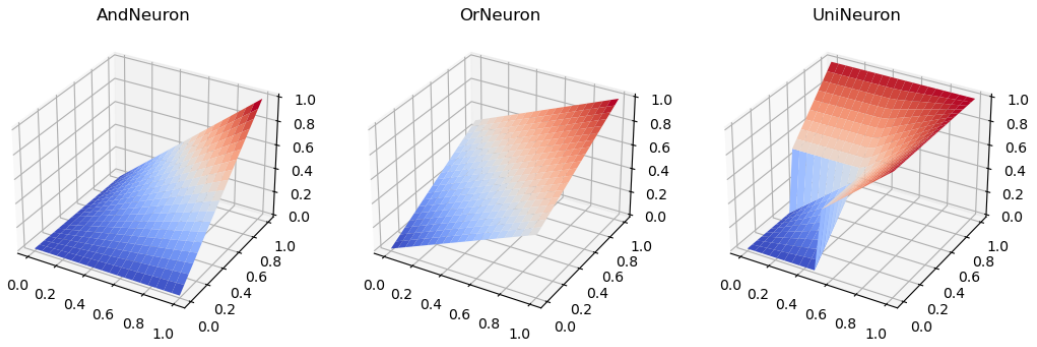


Figure 2. Visualization of fuzzy logic neuron operations: AndNeuron, OrNeuron, and UniNeuron.

2.3.3. Fuzzy Neural Networks

Fuzzy neural networks (FNNs) represent a class of hybrid models that embed fuzzy logic principles into the architecture of conventional neural networks. This integration allows FNNs to effectively manage ambiguity and vagueness in data while retaining the adaptive learning capabilities of neural networks. By combining these complementary strengths, FNNs are particularly well suited for applications that demand both interpretability and resilience in the face of uncertainty. They have been successfully applied in a wide range of contexts, including pattern classification, control engineering, and intelligent decision-making systems, highlighting their relevance in the broader field of artificial intelligence.

In essence, the convergence of fuzzy logic and neural computation has established a robust modeling framework capable of addressing complex, imprecise problems with enhanced flexibility. Positioned at the intersection of symbolic reasoning and data-driven learning, FNNs exemplify the potential of hybrid AI models. As the field continues to progress, FNNs are expected to play a key role in future innovations across areas such as autonomous technologies, healthcare analytics, and real-time decision support systems.

2.4. Innovations and Current Trends in Fuzzy Neural Networks

The year 2023 has witnessed significant advancements in the application of FNNs across various domains, showcasing the versatility and efficiency of integrating fuzzy logic with neural networks and deep learning techniques. A notable trend is the enhancement of performance in specific applications through this integration. For instance, Wang et al. [26] introduced a Residual Gabor Convolutional Network coupled with an innovative data augmentation strategy for finger vein recognition. This approach leverages the characteristics of the Gabor filter to enhance the scale and direction information in pattern features, demonstrating the potential of fuzzy logic in improving feature extraction processes. Similarly, Yu et al. [27] proposed the FS-GAN model, fuzzy self-guided structure retention generative adversarial network, aimed at medical image enhancement. This model particularly excels in handling unpaired data and preserving structural information, illustrating the benefits of fuzzy logic in complex data processing tasks.

In the realm of optimization and efficiency, Li et al. [28] explored neural network controllers' optimization for nonlinear systems, achieving a delicate balance between computational costs and control performance through model compression techniques. This study integrates knowledge distillation and pruning to develop concise neural network-based controllers, underscoring the efficiency of fuzzy logic in optimizing computational resources. Furthermore, Khuat and Gabrys [29] extended the General Fuzzy Min–Max Neural Network to accommodate mixed-attribute data directly, proposing an online learning algorithm that enhances classification performance without requiring encoding techniques for categorical features. This advancement highlights the adaptability and real-time applicability of FNNs in dynamic environments, offering significant improvements in handling mixed-attribute data.

Moreover, the exploration of novel architectures and synchronization techniques within FNN frameworks has opened new avenues for research and application. Hao et al. [30] developed a combined model for wind speed forecasting in urban energy systems, emphasizing the role of mixed frequency modeling and deep learning in improving forecast effectiveness. Additionally, Wang et al. [31] addressed the synchronization of fuzzy neural networks with time-varying delays, focusing on both fixed-time and preassigned-time control strategies. These studies not only showcase the innovative application of FNNs in addressing complex problems but also underscore the continuous evolution of fuzzy logic techniques to enhance the robustness and efficiency of neural network models.

The advancements in 2023 reflect a growing trend toward the integration of fuzzy logic with neural networks and deep learning, leading to the development of more sophisticated and application-specific FNN architectures. These contributions underscore the potential of FNNs in solving complex problems across different domains, promising further innovations and applications in the years to come. Table 1 groups selected studies by their area of application, providing a concise overview of the state of the art in 2023.

Table 1. Summary of fuzzy neural networks and related technologies in 2023.

Area of Application	Approach	Reference
Time-Series Prediction	Self-organizing PRWNN	[32]
Decision Making	Complex Intuitionist Fuzzy	[33]
Robotics	Real-time RNN-based Control	[34]
Fault Detection	Residual Shrinkage Transformer	[35]
Web Services	Swarm Intelligence Search	[36]
Nonlinear Systems Control	Recurrent-Fuzzy System	[37]
Traffic Flow Prediction	Bibliometric Analysis	[38]
Assembly Process	Topic Model-Based NN	[39]
Data Streams	Self-adaptive Fuzzy Learning	[40]
Cybersecurity	Fuzzy Logic-Based Detection	[41]

In 2024, the domain of fuzzy neural networks experienced significant progress, with applications spanning various industries, demonstrating the adaptability and strength of these models. Kan et al. developed an interpretable fuzzy deep neural network (IFDNN) tailored for trading and portfolio rebalancing, employing Moving Average Convergence–Divergence (MACD) indicators and genetic algorithms for the optimization of trading parameters, underscoring the potential of FNNs within financial markets [42]. Concurrently, Singh and Verma delved into aerodynamic modeling, utilizing an Interval Type-3 T-S fuzzy system for nonlinear modeling based on flight data, highlighting the applicability of FNNs in refining aerodynamic assessments [43].

In the burgeoning field of the metaverse, Tavana and Sorooshian conducted a systematic review to assess the role of soft computing methods, including FNN, in shaping this

virtual domain, highlighting the interdisciplinary nature and the necessity for novel developments in FNN applications [44]. Furthermore, Zhao et al. proposed a topology structure optimization for evolutionary hierarchical fuzzy systems, aimed at high-dimensional regression problems, showcasing the adaptability of FNN in dealing with complex data structures [45]. Lastly, Yan et al. addressed the challenge of furnace temperature prediction in municipal solid waste incineration processes through a knowledge transfer online stochastic configuration network, underscoring the effectiveness of FNNs in managing dynamic changes and concept drift in operational conditions [46]. These contributions collectively underscore the dynamic evolution and expanding application spectrum of FNNs, marking 2024 as a year of significant milestones in the field. Table 2 presents some studies to complement this research.

Table 2. Comparative analysis of fuzzy neural network applications in 2023.

Ref.	Problem Area	Main Focus	Architecture	Training Technique
[47]	Robotic Manipulators	Adaptive control	ANFIS with PID	ANFIS estimator feedback
[48]	Regression Problems	Data analysis of complex regression	LFPFC with FCRM clustering	Estimated output-based LFPFC; distance-based LFPFC
[49]	Recommendation Systems	Data embedding with DL and FL	Hierarchical fused neural fuzzy and deep network	Fuzzy-driven HIN embedding
[50]	Financial Time Series	Linear and nonlinear modeling	Cascaded structure with intuitionist fuzzy model	Intuitionistic fuzzy C-means
[51]	Operating Room Performance	Ergonomics in ORs	ANN and DEA	Integrated algorithm using ANN
[52]	Fuzzy Research Analysis	Topic modeling	LDA topic models	Latent Dirichlet allocation (LDA)
[53]	Robot Control	Efficient and precise control	AMDE-BP-FNN	Adaptive and memetic differential evolution with BP

2.5. Advances in Interpretability of Fuzzy Neural Networks

Interpretability in fuzzy systems is paramount for ensuring transparency and trustworthiness in decision-making processes. Recent advancements in fuzzy neural networks (FNNs) have significantly focused on enhancing interpretability and merging the intuitive appeal of fuzzy logic with the powerful learning capabilities of neural networks. This subsection delves into the models of FNNs that seek to enhance interpretability and the main techniques used to evaluate them [54].

FNN, by integrating the principles of fuzzy logic into neural networks, offer a robust mechanism for handling uncertainty and imprecision in data. The interpretability of FNNs is crucial in applications requiring transparent decision making, such as healthcare, finance, and autonomous systems. Several recent studies have focused on improving the interpretability of FNNs through various approaches [55].

Distinguishability (Figure 3) and similarity (Figure 4) are critical for understanding how different inputs are processed and lead to specific outputs. Techniques to enhance these aspects include the use of interpretable fuzzy rules and membership functions. De Campos Souza et al. (2022) demonstrated a novel approach for analyzing the distinguishability of rules in FNNs, leading to more interpretable models [56]. Jim et al. (2024) developed an

approach for pruning and adjusting the ANFIS architecture according to the similarity of fuzzy rules [57].

Completeness in interpretability ensures that the model's explanation covers all aspects of its decision-making process. Efforts to improve completeness often involve the development of comprehensive rule sets that capture the entirety of the model's logic. Pratama et al. [58] developed a fuzzy system model that applies e-completeness criteria to fuzzy rules to select them for the final model. Following is an example of completeness graphical analysis. Figure 5 presents an example of a completeness evaluation.

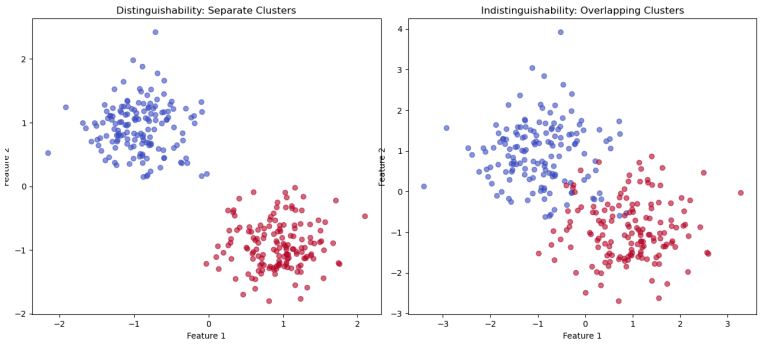


Figure 3. Distinguishable and indistinguishable clusters.

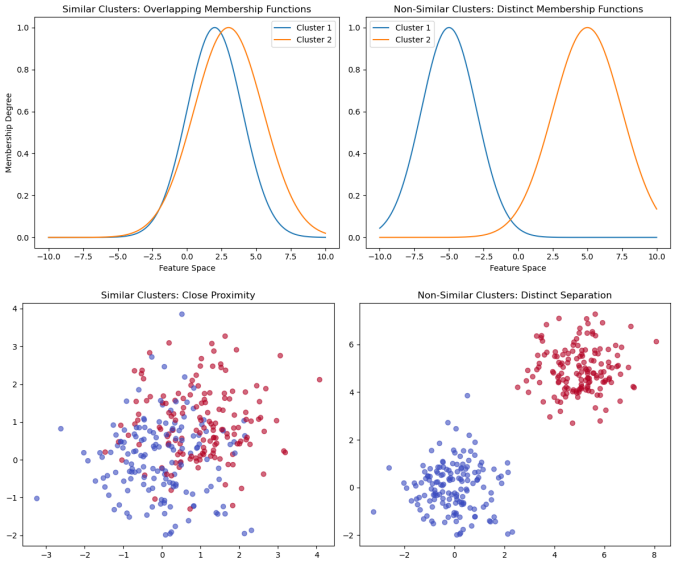


Figure 4. Similarity criteria.

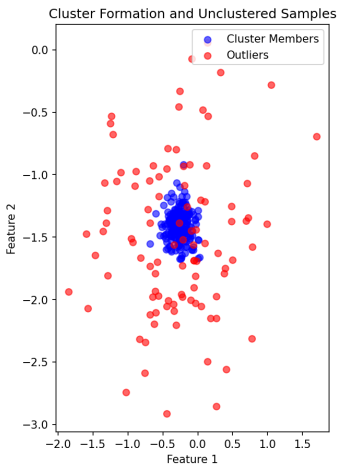


Figure 5. Completeness evaluation.

Feature evaluation enhances interpretability by identifying the significance of each input feature, as shown in Figures 6 and 8. Techniques such as sensitivity analysis and feature importance ranking are commonly employed. Liu et al. (2018) introduced an FNN architecture that incorporates feature evaluation mechanisms, allowing for the direct interpretation of feature contributions to the decision process [59].

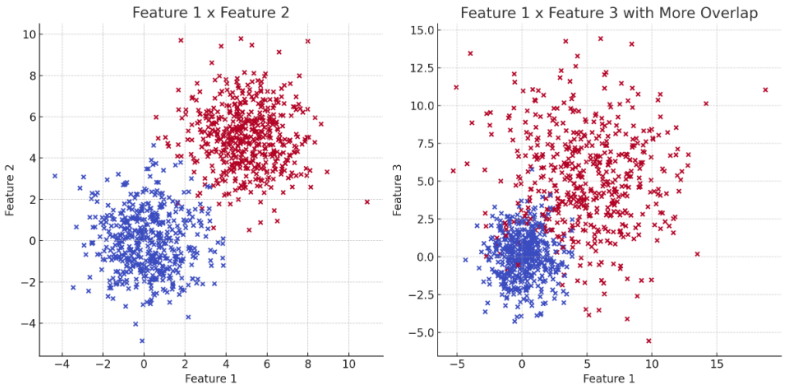


Figure 6. Feature evaluation.

Interpretation of consequents involves understanding the output layer of FNNs, where the fuzzy rules are applied to generate the final decision. Methods to improve this aspect focus on the clarity and simplicity of the rule consequents. De Campos Souza and Lughofer (2022) explored an approach to interpret the consequents in FNNs, making the output decisions more obvious to users [60]. Figure 7 presents an example of consequent interpretation.

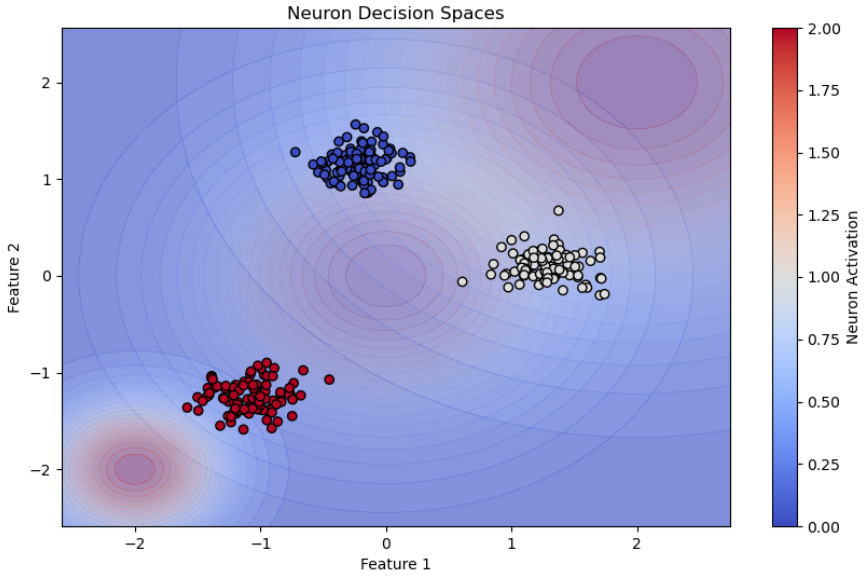


Figure 7. Consequent interpretation: neurons' output for each class.

The quest for interpretable FNNs has led to significant advancements in both the methodologies for developing these networks and the techniques for evaluating their interpretability. As FNNs continue to evolve, the focus on interpretability will ensure their applicability in a wide range of critical decision-making scenarios, fostering trust and transparency in automated systems. Figure 8 presents an example for iris flower detection using an FNN.

Recent works (2022 until now) have leveraged deep learning, reinforcement learning, and fuzzy logic to create systems that not only perform with high accuracy but also provide insights into their decision-making processes. Felizardo et al. (2022) explored the

intersection of machine learning and financial markets through a supervised approach to algorithmic trading in the cryptocurrency market, emphasizing the need for models that offer clear insights alongside performance [61].

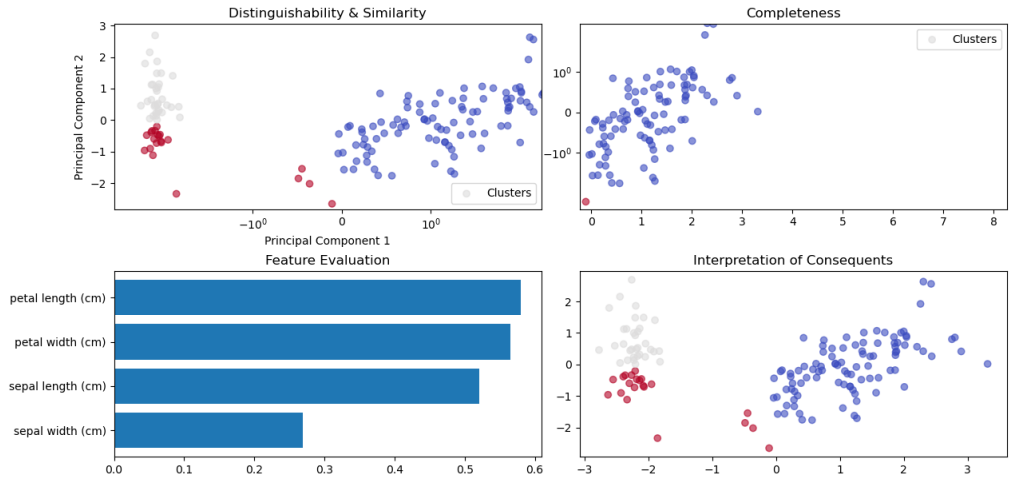


Figure 8. Example of interpretability criteria for Iris dataset.

In the domain of geosciences, Dong et al. (2022) proposed an intelligent resistivity inversion framework based on a fuzzy wavelet neural network (FWNN). This model stands out for its interpretability, facilitated by the integration of the Takagi–Sugeno–Kang (TSK) fuzzy model, which elucidates the rules of inversion results, thereby enhancing the user's understanding of the model's decisions [62].

Furthermore, Hao et al. (2023) developed a novel wind speed forecasting model that combines deep learning strategies with mixed-frequency data processing. This model aims to improve forecast effectiveness for urban energy systems, offering a detailed explanation of its predictive capability, which is crucial for operational planning and low-carbon city construction [30].

Kan et al. (2024) introduced an interpretable fuzzy deep neural network (IFDNN) for trading and portfolio rebalancing. This work emphasizes the importance of interpretability in financial decision making, providing traders with insights into the reasoning behind the network's predictions. The use of fuzzy logic allows for the induction of fuzzy rules from the inference process of neural networks, bridging the gap between human-understandable reasoning and machine learning [42].

Additionally, Hassani et al. (2024) conducted a systematic review of data fusion techniques for optimized structural health monitoring, highlighting the role of interpretability in ensuring the reliability and accuracy of monitoring systems. This review underscores the importance of integrating various data sources to create a coherent understanding of structural health, demonstrating the applicability of interpretability across different domains [63].

Qin et al. (2022) extended work on dropout for the design of TSK fuzzy classifiers, introducing fuzzy rule dropout with dynamic compensation. This approach enhances model generalization and interpretability by mimicking human cognitive behavior more closely [64].

Liao et al. (2023) reimagined multi-criterion decision making by leveraging machine learning methods, focusing on the interpretability of decision-making processes. This literature review identifies the potential of machine learning technologies to uncover patterns and rules from massive datasets, facilitating informed and transparent decision making [65].

Lastly, de Campos Souza and Lughofer (2023) proposed an evolving fuzzy neural classifier that integrates expert rules and uncertainty. This model highlights the importance of interpretability in evolving systems, particularly in contexts where uncertainties can significantly impact decision quality [66].

2.6. Grey Wolf Optimizer (GWO)

The Grey Wolf Optimizer (GWO) is a metaheuristic optimization algorithm inspired by the hunting behavior and leadership hierarchy of gray wolves [67]. The algorithm mimics the natural social structure of gray wolves, which consists of four main roles: alpha (α), beta (β), delta (δ), and omega (ω). These roles influence the decision-making process and hunting strategy used in the optimization process.

The hunting mechanism of the GWO consists of three key phases: encircling prey, hunting, and attacking the prey. These phases are mathematically modeled as follows [67]:

Gray wolves encircle their prey during the hunting process, which is mathematically represented as

$$\vec{D} = |\vec{C} \cdot \vec{X}_p - \vec{X}(t)|, \tag{12}$$

$$\vec{X}(t+1) = \vec{X}_p - \vec{A} \cdot \vec{D}, \tag{13}$$

where

- t is the current iteration;
- $\vec{X}_p$ represents the position of the prey;
- $\vec{X}$ denotes the position vector of a gray wolf;
- $\vec{A}$ and $\vec{C}$ are coefficient vectors computed as

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a}, \tag{14}$$

$$\vec{C} = 2 \cdot \vec{r}_2, \tag{15}$$

  where $\vec{r}_1$ and $\vec{r}_2$ are random vectors in $[0, 1]$, and $\vec{a}$ is a parameter that decreases linearly from 2 to 0 over the course of iterations.

The hunting phase is guided by the best three solutions found so far (i.e., alpha, beta, and delta wolves), and the position of the wolves is updated accordingly [67]:

$$\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}|, \quad \vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}|, \quad \vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}|, \tag{16}$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha, \quad \vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta, \quad \vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot \vec{D}_\delta, \tag{17}$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}. \tag{18}$$

As iterations progress, the value of $\vec{a}$ decreases, reducing the exploration component of the algorithm. This forces the wolves to converge towards the best solution found [67]:

$$\vec{a} = 2 - \frac{2t}{T}, \tag{19}$$

where T is the maximum number of iterations.

The GWO has several advantages that make it an effective optimization technique [67]:

- Simple and easy to implement: It requires few hyper-parameters compared to other swarm-based optimizers.
- Balanced exploration and exploitation: The adaptive parameter $\vec{a}$ helps transition smoothly from exploration to exploitation.

- Global optimization capability: It has been successfully applied to complex, multimodal optimization problems.

GWO has been widely applied in various optimization problems, including [67] the following:

- Feature selection;
- Engineering design problems;
- Neural network training;
- Image processing and computer vision;
- Renewable energy optimization.

3. GWO-FNN: Architecture, Training, and Interpretable Tools

This paper presents the GWO-FNN framework, which improves the interpretability and optimization of FNNs through the integration of the Grey Wolf Optimizer (GWO) and mutual information (MI)-based weight initialization. Building on the work of de Campos Souza [15], which established the foundation for combining fuzzy logic with neural networks, our framework improves pattern recognition by optimizing rule consequent weights with the GWO while ensuring feature relevance through MI-based input weight initialization. These enhancements contribute to a more interpretable and efficient fuzzy neural network for complex classification tasks.

The GWO-FNN framework begins with a fuzzification process that applies Gaussian neurons and uniformly distributed Gaussian membership functions. This transformation converts input features into fuzzy values, establishing a structured foundation for inference.

The intermediate layer consists of logic neurons that implement "AND" and "OR" operations, following the principles introduced by de Campos Souza [15]. This layer enhances the system's ability to identify patterns in data while incorporating improvements suited for modern applications.

A key feature of the framework is the application of the GWO optimization algorithm during training. The GWO optimizes rule-consequent weights while preserving the learned fuzzy rules for post-training analysis. This enables a detailed examination of the rules and the generation of visual representations that reveal the data structure and decision-making processes.

The GWO-FNN framework retains all fuzzy rules after training, ensuring greater model interpretability, and integrating fuzzy logic and neural network methodologies, with GWO playing a central role in refining the rule consequents and improving classification performance.

3.1. Optimization Methods Applied to Fuzzy Neural Networks

Fuzzy neural networks (FNNs) are widely recognized for their strong problem-solving capacity, especially when incorporating interpretability aspects. The more finely tuned the network parameters, the better the responses obtained. Various optimization techniques have been applied to enhance parameter tuning, architectural definitions, and overall network performance.

Several studies have focused on structural and parametric optimization to improve both accuracy and interpretability in fuzzy neural models:

- Structural optimization: Pizzileo et al. [68] proposed an approach that simultaneously optimizes the number of inputs and rules, ensuring a balance between interpretability and accuracy.
- Multi-objective evolutionary algorithms: Gómez-Skarmeta et al. [69] introduced a multi-objective evolutionary algorithm that considers both accuracy and interpretability criteria to generate optimal fuzzy models.

- Gradient-based learning: Zhao et al. [70] developed a gradient descent approach that optimizes premise and consequent parameters simultaneously, improving both interpretability and accuracy.

Metaheuristic techniques, such as genetic algorithms (GAs), particle swarm optimization (PSO), and the Grey Wolf Optimizer (GWO), have been employed to enhance the tuning of fuzzy neural networks:

- Genetic algorithms (GAs): Genetic optimization has been used to refine fuzzy rules and parameters [71].
- Fuzzy rough neural networks: Cao et al. [72] introduced evolutionary fuzzy rough networks that optimize both interpretability and predictive performance.
- Levenberg–Marquardt-optimized fuzzy models: Ebadzadeh and Salimi-Badr [73] implemented hierarchical Levenberg–Marquardt optimization for function approximation.

The Grey Wolf Optimizer (GWO) is an emerging metaheuristic inspired by the hunting behavior of gray wolves, which has been used in fuzzy neural networks for parameter optimization and architectural tuning.

- Fuzzy GMDH neural network optimized by GWO: Heydari et al. [74] applied the GWO to optimize a fuzzy group method of data handling (GMDH) neural network for wind turbine power forecasting, leading to higher accuracy in energy predictions.
- Modified GWO for learning rate selection in fuzzy controllers: Le et al. [75] proposed a modified GWO to fine-tune learning rates in a multilayer fuzzy controller, improving both convergence and system stability.
- Fuzzy strategy GWO for multimodal problems: Qin et al. [76] developed a fuzzy strategy GWO (FSGWO) for multimodal optimization, demonstrating superior convergence over traditional GWOs.
- GWO in modular granular neural networks: Sánchez et al. [77] applied GWO to optimize modular neural networks for biometric recognition, achieving significant improvements in accuracy.

The literature confirms that fuzzy neural networks benefit significantly from optimization techniques aimed at improving accuracy and interpretability. Methods such as multi-objective evolutionary algorithms, genetic algorithms, and Levenberg–Marquardt optimization have been widely adopted.

The GWO has emerged as a promising technique, providing enhanced parameter tuning capabilities for fuzzy neural networks. Applications of the GWO in fuzzy systems demonstrate the following:

- Improved forecasting performance;
- Better interpretability–accuracy trade-offs;
- More efficient learning rate adaptation.

Given the success of the GWO in optimizing fuzzy neural networks, its use for parameter optimization represents an innovative contribution to the field.

3.2. Fuzzy Neural Network Architecture: Structure of Variable Neurons and Activation Functions

The GWO-FNN framework initiates its processing with a fuzzification layer comprising L neurons, each employing equally spaced membership functions instead of deriving them from input data density. This structure allows for defining L membership clouds (A_{lj}) for each input feature x_{ij}. These membership functions act as activation functions, generating membership degrees ($a_{jl} = \mu_l^A$) for $j = 1 \ldots N$ and $l = 1 \ldots L$, with a grid partition algorithm determining the number of inputs (N) and fuzzy sets per input (L).

The rule formation layer consists of L logical neurons responsible for processing fuzzy rule antecedents. These neurons implement operations such as "AND" (Equation (8)), "OR"

(Equation (9)), and "UNI" (Equation (11)) logic. Each neuron aggregates outputs from the fuzzification layer using randomly assigned feature weights (w_{il}) for $i = 1 \ldots N$ and $l = 1 \ldots L$. The selection of a single activation value a_{jl} per input variable j ensures that the extracted fuzzy rules effectively represent the dataset.

The defuzzification layer employs an aggregation neuron utilizing the Sigmoid activation function:

$$f_\Gamma(z) = \frac{1}{1 + e^{-z}} \tag{20}$$

This transition from fuzzy to crisp outputs allows the GWO-FNN to handle classification tasks. The final output is determined using the Ω function, which maps results based on a thresholding mechanism:

$$\Omega = \begin{cases} 1, & \text{if } \sum_{j=0}^{l} f_\Gamma(z_j, v_j) > 0 \\ 0, & \text{if } \sum_{j=0}^{l} f_\Gamma(z_j, v_j) \leq 0 \end{cases} \tag{21}$$

By incorporating the Grey Wolf Optimizer (GWO), our model enhances rule-consequent tuning, ensuring that fuzzy rules remain interpretable while optimizing classification accuracy. As described in Section 2.6, the GWO mimics the hierarchical decision-making and hunting strategies of gray wolves to explore and exploit the search space effectively. In our implementation, GWO utilizes a pack of P_n wolves, where P_n is determined through cross-validation to optimize the balance between exploration and exploitation. Inspired by the work of de Campos Souza [78], this structured approach refines fuzzification, rule construction, and defuzzification, offering an improved framework for complex pattern recognition tasks.

The fuzzy rules can be extracted from the network topology in the form as presented in Equation (22) below:

$$\begin{aligned} Rule_1: \text{ If } x_1 \textit{ is } A_1^1 \text{ with impact } w_{11} \ldots \\ AND/OR_{(g)}\ x_2 \textit{ is } A_1^2 \text{ with impact } w_{21} \ldots \\ \text{Then } y_1 \textit{ is } \uplus_1 \\ Rule_2: \textit{ If } x_1 \textit{ is } A_2^1 \text{ with impact } w_{12} \ldots \\ AND/OR_{(g)}\ x_2 \textit{ is } A_2^2 \text{ with impact } w_{22} \ldots \\ \text{Then } y_2 \textit{ is } \uplus_2 \\ \ldots \\ Rule_L: \text{ If } x_1 \textit{ is } A_L^1 \text{ with impact } w_{1L} \ldots \\ AND/OR_{(g)}\ x_2 \textit{ is } A_L^2 \text{ with impact } w_{2L} \ldots \\ \text{Then } y_L \textit{ is } \uplus_L \end{aligned} \tag{22}$$

with $\uplus$ representing the outputs of the fuzzy rule consequents, and A denoting the Gaussian neuron (fuzzy set) generated in the first layer (Equation (23)). The model employs $v_1, v_2, \ldots, v_L$ for binary classification problems [60]. The parameter g, derived from uninorms, is randomly initialized within the range [0, 1], determining whether the system operates as an AndNeuron (AND connective) or an OrNeuron (OR connective). This parameter plays a crucial role in defining rule groups, explicitly distinguishing between AndNeuron and OrNeuron connections. Within the uninorm framework, the operational mode—favoring either AND or OR connectivity—is governed by the assigned value of g. This configuration allows for a structured yet adaptable approach to rule formation, enhancing the model's interpretability and classification capability across complex datasets.

For binary classification, the consequents $v_1, \ldots, v_L$ take singleton values, each corresponding to a specific class label.

Furthermore, the weight term w_{il} quantifies the contribution of the ith feature in the lth rule, influencing the activation level of fuzzy neurons in the second layer.

Figure 9 depicts the feed-forward architecture of the implemented FNN, illustrating the fuzzification stage in the first layer, the development of fuzzy neurons in the second layer, and the aggregation neuron in the final layer. The synergy between the first two layers forms a neuro-fuzzy inference system capable of constructing fuzzy rules that extract valuable knowledge from the examined dataset.

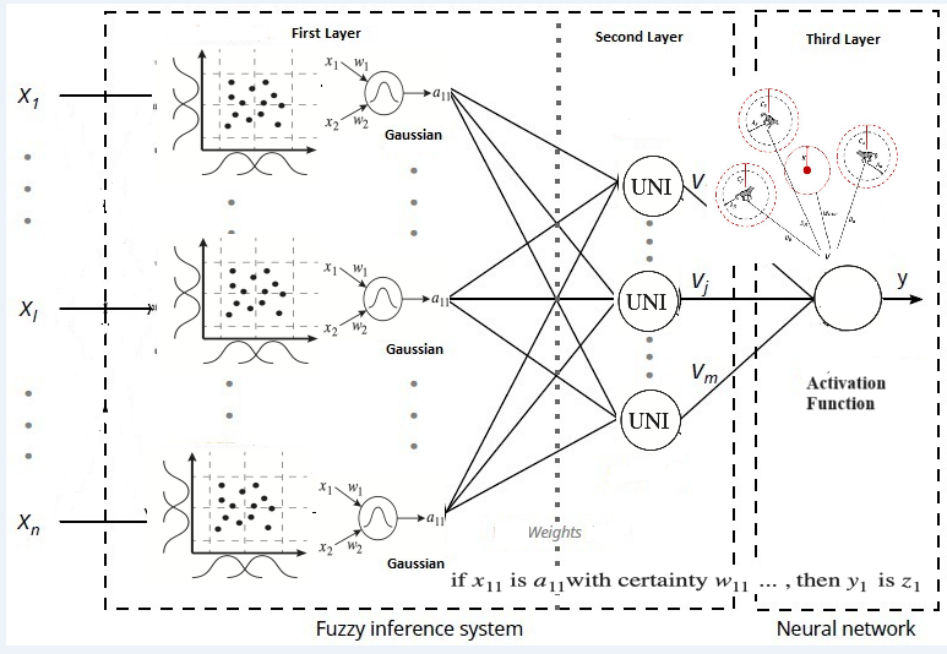


Figure 9. FNN architecture.

The weights of the output neural network, optimized using the GWO technique, represent the conclusions of the fuzzy inference system. The rules extracted by the first two layers define how different features contribute to the predicted outcomes. This process highlights the importance of feature integration in shaping the expected results, achieved through the combined application of neural network optimization and fuzzy logic inference.

3.3. First Layer: Grid-Based Fuzzification with Uniform Membership Functions

The initial layer of the FNN model employs a grid-based fuzzification strategy, which plays a pivotal role in transforming raw input data into structured fuzzy representations. This method ensures a uniform distribution of membership functions across each input domain, laying the groundwork for systematic and interpretable fuzzy rule construction [79].

In this layer, each input variable x_j is mapped to a set of M fuzzy sets A_{lj}, which act as activation functions for the corresponding neurons. These functions return the degree to which an input value belongs to a particular fuzzy set and are defined mathematically as [79]

$$a_{jl} = \mu_l^A(x_j), \quad \text{for } l = 1, \ldots, M. \tag{23}$$

The fuzzy sets are derived through a uniform grid partitioning of the input space, where each partition segment is modeled using a Gaussian membership function. Each Gaussian neuron is parameterized by a center c_{jl} and a standard deviation σ_{jl}, both determined based on the distribution of the input data [79].

The membership function, representing the activation level of input x_j relative to the lth fuzzy set of the jth variable, is expressed as

$$\omega(x_j, c_{jl}, \sigma_{jl}) = e^{-\frac{1}{2}\left(\frac{x_j - c_{jl}}{\sigma_{jl}}\right)^2}, \quad \text{for } j = 1, \ldots, N, \quad l = 1, \ldots, M, \tag{24}$$

where N denotes the total number of input variables, and M corresponds to the number of fuzzy sets per input. The parameters c_{jl} and σ_{jl} define the center and spread of each Gaussian function, enabling a smooth and interpretable mapping of inputs to fuzzy activations.

To ensure that the initial weights of Gaussian neurons reflect the relevance of each input feature in classification tasks, we employ a mutual information (MI) approach. The MI between a feature x_j and the class label y is given by

$$I(x_j, y) = \sum_{x_j \in X} \sum_{y \in Y} P(x_j, y) \log \frac{P(x_j, y)}{P(x_j)P(y)}, \tag{25}$$

where $P(x_j, y)$ represents the joint probability distribution of x_j and y, while $P(x_j)$ and $P(y)$ denote their marginal probabilities.

The MI values are normalized within a predefined range $[w_{min}, w_{max}]$ to define the initial weights:

$$w_j = w_{min} + (w_{max} - w_{min}) \frac{I(x_j, y) - \min(I)}{\max(I) - \min(I)}. \tag{26}$$

This guarantees that features with higher discriminative power receive higher initial weights, while less relevant features contribute minimally to the decision process.

This detailed exposition of the grid-based fuzzification process with uniform membership functions underscores the mathematical precision employed to convert input data into fuzzy sets effectively, laying a solid foundation for the systematic generation and interpretation of fuzzy rules within the FNN model.

The adoption of equally spaced membership functions greatly enhances the visualization clarity of the analyzed patterns, which shows how varying the number of partitions can alter the problem interpretation.

Within the sphere of FNNs employing a grid-based fuzzification methodology, the introduction of complexity by this approach, especially under particular conditions, warrants careful consideration. Crucially, the interaction between the number of features (N) and the number of membership functions (M) per feature precipitates an exponential increase in the creation of neurons. This proliferation is mathematically represented by the relationship $L = M^N$, where L denotes the total number of neurons generated. Such exponential growth underscores the intricate relationship between feature quantity and model complexity, alongside implications for interpretability.

A moderate expansion in model partitioning subdivisions may indeed bolster accuracy but at the potential expense of diminishing the network's interpretability. In fuzzy system design and evaluation, interpretability is paramount, urging a judicious approach to how the model's partitioning is determined.

Referencing seminal research on human cognitive abilities and the integration of features within a rule-centric framework, it has been posited that the optimal threshold for comprehensive understanding does not exceed seven dimensions. This cap, frequently associated with the cognitive constraints delineated in Miller's (1956) [80] seminal exposition on the magical number seven, plus or minus two, highlights the finite capacity of human cognition to process and interpret multidimensional information. Pertinently, within fuzzy system discourse, this cognitive ceiling accentuates the critical need for balancing

model intricacy against human interpretability, facilitating effective system utilization and insight acquisition.

Therefore, the formulation of FNNs underpinned by grid-based fuzzification demands an intentional strategy regarding feature selection and partitioning. This strategy aims to harmonize model precision with interpretability, adhering to established cognitive limits on dimensionality. Such adherence not only augments FNN usability but also aligns with the broader ambition of cultivating intuitive and transparent decision-making frameworks.

3.4. Second Layer: Fuzzy Rule Extraction

The model's second layer has fuzzy neurons capable of aggregating the fuzzy Gaussian neurons generated in the first layer along with their respective weights. This union is calculated using the concepts expressed in Equation (8), Equation (9), or Equation (11), where they aggregate the weights with the Gaussian value and transform it into a single value, which comprises the rule antecedents connected with the respective connective correlated to the respective neuron used. This type of mechanism is the first step towards building an expert system based on fuzzy rules generated from the data. This facilitates a real understanding of facts and elements present in the data that often go unnoticed by the human eye.

After compiling the fuzzy rule antecedents, completing these rules with their respective consequents is necessary. These are factors that define how much a fuzzy rule reveals about the class that it is best able to identify. The rule consequent is calculated according to the input data and based on the classes of the problem involved.

3.5. Third Layer: Neural Aggregation Output Layer

The third and final layer of the fuzzy neural network (FNN) architecture is responsible for executing the defuzzification process, serving as the bridge between fuzzy reasoning and crisp output generation. This layer aggregates the responses from the fuzzy neurons in the second layer to produce a single, interpretable output value. The aggregation mechanism is governed by a well-defined mathematical formulation, which encapsulates the contribution of each fuzzy rule and facilitates the transformation from fuzzy logic to a precise numerical result. The corresponding expression can be expanded to offer a clearer and more detailed interpretation of the output computation:

$$\hat{y} = \Omega\left(\Gamma\left(v_0 + \sum_{j=1}^{L} z_j \cdot v_j\right)\right), \tag{27}$$

wherein Γ signifies the Sigmoid activation function (Equation (20)) and Ω is Equation (21), designed to confine the aggregated neural inputs within a bounded range of (0, 1). This feature renders the Sigmoid function particularly advantageous for binary classification endeavors and the elucidation of fuzzy logic rules.

In the aggregation formula, $z_0 = 1$ serves as a bias term, integrating a constant offset v_0 (the bias weight) into the equation to adjust the activation threshold. The terms z_j and v_j for $j = 1, \ldots, L$ correspond to the outputs from the second layer's fuzzy neurons and their associated weights, respectively. The inclusion of the bias term ensures that the neural network can effectively handle scenarios where the decision boundary does not intersect the origin, thereby enhancing the model's flexibility and adaptability.

The use of the Sigmoid activation function in the aggregation equation plays a crucial role in mapping the linear combination of fuzzy neuron outputs and their weights to a probability space. This mapping facilitates the final interpretation of the aggregated outputs, seamlessly bridging the gap between the fuzzy inference system and the production of a definitive output $\hat{y}$, which represents the model's prediction.

The third layer effectively encapsulates the defuzzification process, transforming fuzzy inferences into precise, actionable insights. The choice of the Sigmoid function underscores the layer's ability to interpret fuzzy logic within a probabilistic framework, culminating in a robust mechanism for the conclusive interpretation of complex patterns discerned by the FNN.

The defuzzification mechanism employed within the model is meticulously shown in Figure 10. This illustration delineates the transition of Z values from a state of fuzzy ambiguity to quantifiable certainties, mapping them into a definitive range between 0 and 1. Observing this figure, one can pinpoint the precise moments where aggregated outputs, represented by Z values, intersect with the Sigmoid activation function. These intersections are pivotal, as they mark the conversion points where fuzzy inputs are defuzzified into crisp outputs.

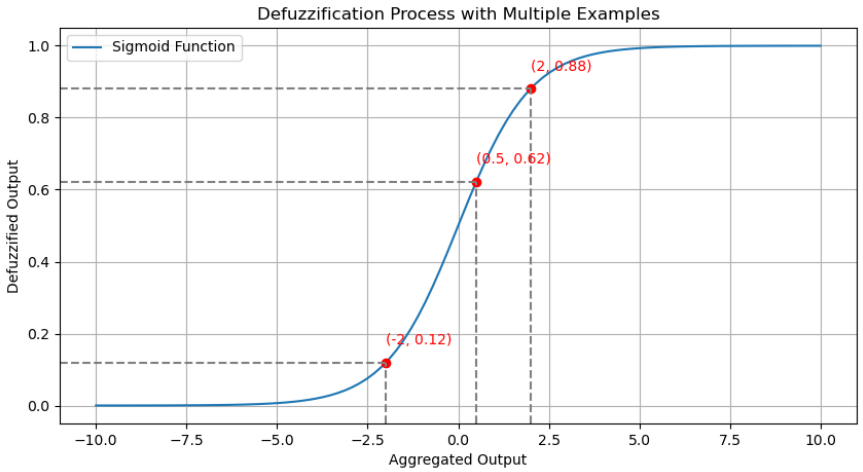


Figure 10. Defuzzification process using Sigmoid function.

Particularly noteworthy in Figure 10 are the projection lines extending from the points of intersection down to the horizontal axis and up to the vertical axis. These lines serve a dual purpose: firstly, they highlight the original aggregated output values on the Z axis before defuzzification; and secondly, they reveal the corresponding defuzzified outputs, now transformed and residing within the binary continuum. The inclusion of multiple examples with distinct Z values, each being mapped through the Sigmoid function, enriches the reader's understanding by demonstrating the broad applicability and robustness of the defuzzification process across varying degrees of input data.

Moreover, the selected examples are strategically distributed along the Sigmoid curve, encompassing cases with negative, neutral, and positive aggregated outputs. This diversity highlights the Sigmoid function's ability to accommodate a broad spectrum of input values and reinforces its importance in the defuzzification process. By producing outputs with inherently probabilistic characteristics, the Sigmoid function supports decision making that is statistically grounded and interpretable.

3.6. Training of the Third Layer with Grey Wolf Optimization (GWO)

The training of the model's third layer is conducted using the Grey Wolf Optimization (GWO) algorithm [67], an efficient nature-inspired metaheuristic that mimics the hierarchical hunting mechanism of gray wolves in the wild. The GWO is particularly advantageous in global optimization tasks, as it effectively balances exploration and exploitation, thereby avoiding local minima.

In the context of the GWO-FNN model, the GWO refines the weights of neurons, including AndNeurons, OrNeurons, and UniNeurons (as detailed in Equation (8),

Equation (9) and Equation (11), respectively), through an iterative process based on the social hierarchy of alpha, beta, delta, and omega wolves. The weight update mechanism in the GWO follows the following rule:

$$\vec{v}_k^{(t+1)} = \vec{v}_\alpha^{(t)} - A \cdot \left| D \cdot \vec{v}_\alpha^{(t)} - \vec{v}_k^{(t)} \right| \tag{28}$$

where

- $\vec{v}_\alpha$ represents the best solution found so far (alpha wolf position);
- A is the adaptive control parameter, defined as $A = 2a \cdot r_1 - a$, where a linearly decreases from 2 to 0 over a number of iterations;
- D represents the distance between the candidate solution and the leader, given by $D = |C \cdot \vec{v}_\alpha - \vec{v}_k|$, where $C = 2 \cdot r_2$;
- r_1, r_2 are random values in the range $[0, 1]$ for stochastic behavior.

The GWO approach enables the iterative adjustment of weights, refining the optimization process based on the collective intelligence of the gray wolf pack. This adaptive optimization mechanism ensures improved convergence towards optimal solutions across various neural network architectures.

For our GWO-FNN model, the final weight optimization of the last layer is expressed as

$$\vec{v}_k = \text{GWOOptimization}(\vec{Z}, \vec{y}_k), \quad \forall k = 1, \ldots, C \tag{29}$$

where C denotes the number of classes, $\vec{Z}$ contains the activation outputs of all neurons for input samples, and $\vec{y}_k$ is the corresponding target vector for binary classification, aligning with the Sigmoid function's output.

The optimization process in GWO is guided by a fitness function that evaluates the quality of the consequent weights $\vec{v}_k$ based on the classification performance. In this study, we define the fitness function as the mean squared error (MSE) between the predicted outputs and the actual target values:

$$\text{Fitness}(\vec{v}) = \frac{1}{N} \sum_{i=1}^{N} (y_i - f_\Gamma(z_i))^2, \tag{30}$$

where

- N is the total number of training samples;
- y_i is the actual class label for sample i;
- $f_\Gamma(z_i)$ represents the network's output after applying the Sigmoid activation function.

The GWO algorithm seeks to minimize this function by adjusting the consequent weights, ensuring that the fuzzy rule set provides optimal classification performance. This fitness evaluation is performed iteratively for each wolf in the pack, with the alpha, beta, and delta wolves leading the optimization process toward a lower-error solution.

The application of the GWO in fuzzy neural networks improves the interpretability and robustness of the model by refining the weights based on a global search strategy. An example of optimization using the GWO is illustrated in Figure 11.

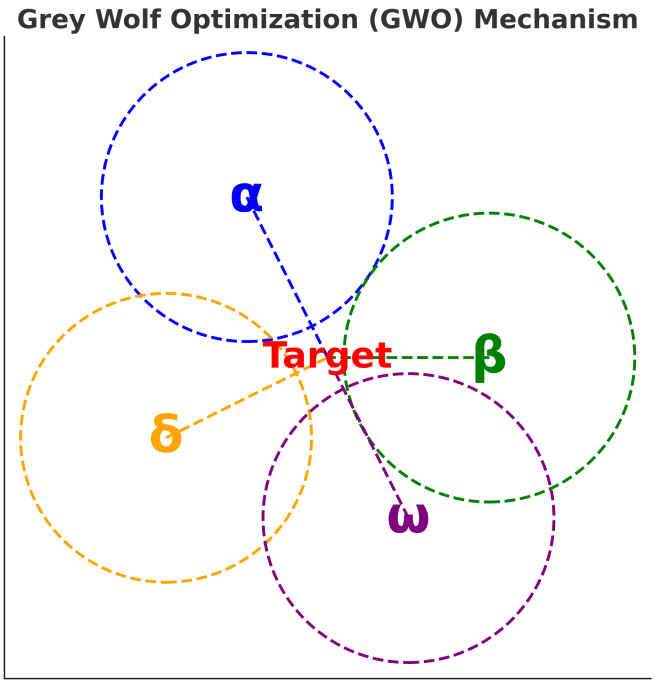


Figure 11. GWO optimization.

Figure 11 illustrates the Grey Wolf Optimization (GWO) mechanism, which simulates the hierarchical structure of gray wolves during hunting. The alpha (α), beta (β), and delta (δ) wolves guide the search, while the omega (ω) wolves explore new areas. The target represents the optimal solution, towards which the wolves adjust their positions iteratively. The dashed lines indicate the movement direction, and the circles represent the influence zone of each wolf, aiding in convergence toward the best solution.

The GWO-FNN model has two primary parameters: the grid partition (γ) for the fuzzification method, and the number of wolves in the Grey Wolf Optimizer (P_n). The training process utilizes the GWO to optimize the consequent weights of fuzzy rules instead of traditional gradient-based methods. The complete training process is detailed in Algorithm 1 and a schematic figure (Figure 12) presents the main steps involved in the training process.

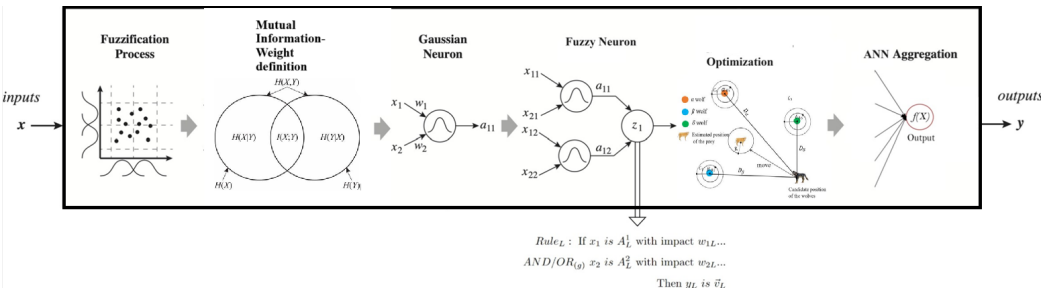


Figure 12. Fuzzy neural network training: main steps.

Algorithm 1 Fuzzy Neural Network Training Process using Grey Wolf Optimizer (GWO)

Input: Dataset $\vec{X}$, target vector $\vec{y}$, number of wolves P_n.
Step 1: Initialize grid partition parameters and the number of membership functions (*num_mfs*) for each input feature.
Step 2: Uniformly calculate centers (c_{jl}) and standard deviations (σ_{jl}) for L fuzzy sets per input feature.
Step 3: Apply Gaussian membership functions to construct L fuzzy neurons for each feature, as defined by Equation (23).
Step 4: Generate fuzzy neuron outputs for each input sample using the Gaussian functions, forming $\vec{Z}$.
Step 5: Adapt target vector $\vec{y}$ for binary classification tasks.
Step 6: Initialize the Grey Wolf Optimizer (GWO) with P_n wolves and set the stopping criteria.
Step 7: Encode the consequent weights $\vec{v}$ as the position of wolves in the search space.
Step 8: Compute the fitness of each wolf based on classification accuracy.
Step 9: Update the positions of the alpha (α), beta (β), and delta (δ) wolves following the GWO update equations (Section 2.6).
Step 10: Continue updating until the stopping criteria are met (e.g., max iterations or convergence).
Step 11: Select the best solution (position of the α wolf), which represents the optimized consequent weights $\vec{v}^*$.
Output: FNN trained with optimized fuzzy rules.

4. Advanced Interpretation of the Rules Extracted from the Evolving Neurons

Enabling the model's architecture and operation to be interpretable facilitates human understanding and acceptance of computational techniques. In this section, methods for evaluating the interpretability of the model proposed in this paper are presented.

Auxiliary Methods for Interpretability

Interpretability in fuzzy neural networks extends beyond the initial fuzzification layer. Unlike traditional black-box neural networks, FNNs provide an additional layer of transparency by structuring the second layer as a fuzzy rule inference mechanism. To further enhance interpretability, we employ two advanced visualization techniques:

- Heatmap visualization of logic neuron outputs: This technique provides a clear understanding of how different logical neurons contribute to the final classification decision. The intensity of each neuron's activation is mapped to a color scale, revealing patterns in neuron activations across different samples.
- Three-dimensional projection of logical neuron outputs: By visualizing the response of fuzzy logic neurons in a three-dimensional space, we can analyze how input features influence decision making in a nonlinear fashion. This approach is particularly useful for understanding how different membership functions interact to generate final predictions.

These methods ensure that, even with multiple layers, our FNN model remains interpretable. The second layer directly corresponds to human-understandable fuzzy rules, and the third layer aggregates these rules while maintaining transparency in decision making.

5. Computational Complexity Analysis with Parameter Nomenclature

The GWO-FNN algorithm comprises multiple processing stages, each contributing uniquely to its overall computational demand. This section outlines the complexity analysis for each component, using the parameters introduced earlier to ensure consistency and clarity.

Fuzzification layer: In this initial phase, each input vector $\vec{x}$ is mapped to a set of fuzzy activations $\vec{a}$ through Gaussian membership functions defined by centers $\vec{c}_i$ and standard deviations $\vec{\sigma}_i$. For a dataset containing n samples, N input features, and k fuzzy sets per feature, the complexity of computing these activations is $O(n \cdot N \cdot k)$.

Logical neuron layer: The outputs of the fuzzification layer are processed through logic neurons that apply aggregation functions such as t-norms and t-conorms. The number of neurons formed is $L = k^N$, representing all possible fuzzy rule combinations. Consequently, this layer has a complexity of $O(n \cdot k^N)$, reflecting its exponential dependence on the number of inputs.

Training phase with GWO: Optimization using the Grey Wolf Optimizer (GWO) involves population-based search governed by interactions among the top candidate solutions—commonly referred to as the alpha, beta, and delta wolves. Let e be the number of epochs, b the batch size, and q the cost of processing one batch. The resulting training complexity is $O(e \cdot \frac{n}{b} \cdot q)$. Unlike gradient descent methods, GWO updates are position-based and do not rely on explicit bias terms, with computational load influenced by the number of wolves in the population.

Model evaluation: To assess the model's predictive performance, comparisons are made between predicted outputs $\hat{y}$ and true labels y_k across n_{test} testing samples. This operation scales linearly with the test set size, yielding a complexity of $O(n_{test})$.

Interpretability assessment: Evaluating interpretability involves computing similarity and consistency matrices, as well as the ϵ-completeness metric ϵ-Completeness(X, R, ϵ). Letting r represent the number of logic neurons, the complexity for these computations is approximately $O(r^2)$, due to pairwise comparisons within the rule base.

Total complexity: Integrating all components, the overall computational cost of the GWO-FNN framework can be approximated as

$$O(n \cdot N \cdot k) + O(n \cdot k^N) + O(e \cdot \frac{n}{b} \cdot q) + O(n_{test}) + O(r^2). \tag{31}$$

This formulation illustrates the algorithm's computational footprint and highlights the trade-offs between model complexity, interpretability, and training efficiency in real-world applications.

6. Experimental Evaluation

To assess the effectiveness and interpretability of the proposed GWO-FNN model, a series of experiments were conducted. Rather than relying solely on performance metrics, the evaluation emphasizes the model's capacity to provide interpretable reasoning in binary classification tasks. The experimental design is organized into three parts, each focusing on a different aspect of model behavior and interpretability.

6.1. Understanding Interpretability Using Synthetic Data

The first phase of the evaluation employs a synthetic dataset specifically crafted to test how the model reveals its internal decision mechanisms. Synthetic data are ideal for this purpose, as they offer full control over the input–output relationships, enabling a precise analysis of how the model processes information and how its outputs align with logical expectations.

This setup allows us to directly examine the structure and function of fuzzy rules generated by the network. Key interpretability tools used include the visualization of logic neuron activations, 3D projection of neuron outputs for insight into the rule space, and similarity matrices to evaluate the consistency of the fuzzy rules. These techniques help uncover the decision patterns embedded in the model.

Multiple configurations of the model are explored, varying both the neuron type (e.g., AndNeuron, OrNeuron, and UniNeuron) and the number of membership functions per input. This comparative approach allows us to investigate how different architectural choices influence not only classification performance but also the clarity and interpretability of the learned rules.

6.2. Comparative Analysis with State-of-the-Art Models

Following the interpretability exploration, the second subsection transitions to a comparative study, positioning the proposed model against contemporary state-of-the-art models in the domain of binary pattern classification. This comparative analysis aims to highlight the model's relative performance, accuracy, and, critically, its interpretability vis-à-vis existing methodologies. By benchmarking against established models, this section endeavors to contextualize the proposed model's efficacy and its contribution to advancing interpretability in binary classification tasks.

This study meticulously constructs an experimental framework to critically assess the GWO-FNN model against established machine learning paradigms, including Multilayer Perceptron (MLP), Naive Bayes, Random Forest, among others. The strategic selection of these comparative models provides a holistic view across traditional and ensemble learning methodologies, enabling a comprehensive performance evaluation of the GWO-FNN model's classification capabilities.

The datasets employed in this experiment were carefully chosen to span a broad spectrum of applications, ensuring the robustness and applicability of the evaluation. These datasets, listed below (Table 3), originate from the UCI Machine Learning Repository and have undergone standardized pre-processing techniques such as normalization and categorical encoding.

Table 3. Summary of datasets used in this study, including the number of samples and features.

Dataset	Samples	Features
Iris Dataset [81]	150	4
Mammographic Masses [82]	961	5
Haberman's Survival [83]	306	3
Blood Transfusion Service Center [84]	748	4
Liver Disorders [85]	345	6
Immunotherapy Dataset [86]	90	7
Cryotherapy Dataset [86]	90	7

Central to the GWO-FNN model's evaluation is its innovative tri-layered architecture, where logical neurons in the intermediary layer are adeptly transformed into definitive fuzzy rules. This mechanism significantly enhances the model's interpretability by offering a transparent view into its reasoning process. The assessment of model performance is grounded on a range of metrics, specifically the following:

- Accuracy: This is defined as the proportion of true results (both true positives and true negatives) among the total number of cases examined. It can be mathematically represented as $\frac{TP+TN}{TP+FN+TN+FP}$.
- Precision: This metric reflects the proportion of true positive results in all positive predictions made by the model, given by $\frac{TP}{TP+FP}$.
- Recall: Also known as sensitivity, this measures the proportion of actual positives that are correctly identified, calculated as $\frac{TP}{TP+FN}$.
- F1 score: The F1 score is the harmonic mean of precision and recall, providing a balance between the two. It is computed as $2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$.

Where TP denotes true positive, TN represents true negative, FN stands for false negative, and FP indicates false positive. These metrics collectively provide a nuanced insight into the efficacy of the model in classifying the datasets under study.

- GWO-FNN variants: We evaluated three versions of the proposed model—GWO-FNN with AndNeuron, OrNeuron, and UniNeuron logic configurations. Each variant utilized four membership functions per input dimension, selected based on preliminary experiments comparing different settings (mf = [2, 3, 4, 5]). The Adam optimizer was incorporated to enhance convergence during training.
- Baseline models: As benchmarks, we selected well-established classifiers. The Multilayer Perceptron (MLP) [87] was implemented with a single hidden layer containing 100 neurons to ensure sufficient learning capacity. The Random Forest Classifier [88] employed 100 decision trees to balance predictive accuracy and overfitting resistance. Lastly, the Gaussian Naive Bayes classifier [89] was used with default parameters, offering a probabilistic baseline.

All models were trained and evaluated using a 70:30 train–test split, a common practice in supervised learning to ensure generalization capability. Our goal was to determine whether the GWO-FNN variants could match or outperform conventional algorithms based on standard classification metrics: accuracy, precision, recall, and F1 score.

To ensure statistical robustness, each experiment was repeated 30 times, allowing us to assess the stability and consistency of each model's performance. We conducted a one-way analysis of variance (ANOVA) [90] on the collected accuracy results to identify statistically significant differences among the models.

In cases where ANOVA indicated significance, we followed up with Tukey's honestly significant difference (HSD) test [91] to identify specific model pairs with meaningful performance differences. This post hoc analysis was essential for interpreting the relative strengths of each classifier.

To validate the assumptions required for the ANOVA, we tested the normality of residuals using the Shapiro–Wilk test [92], and evaluated the homogeneity of variances across groups via Levene's test [93]. These checks ensured that the statistical conclusions drawn were both valid and reliable.

By integrating this rigorous evaluation pipeline, our study not only benchmarks the GWO-FNN variants against standard models but also provides a statistically grounded perspective on their practical effectiveness. This methodological rigor reinforces the credibility of the results and supports the GWO-FNN's potential as a viable solution for interpretable classification tasks.

6.3. Application to a Real-World Academic Dataset: Sepsis Dataset

In the final phase of our experimental evaluation, we assess the performance of the proposed GWO-FNN models using a real-world dataset focused on sepsis—a critical and time-sensitive medical condition. This stage is designed to move beyond controlled, synthetic scenarios and evaluate the model's ability to operate effectively in complex, high-stakes environments such as healthcare.

The use of the sepsis dataset, derived from well-established research, serves a dual purpose: first, to validate the model's adaptability to real clinical data; and second, to investigate whether the interpretable nature of GWO-FNN can provide actionable insights in medical decision making. Given the urgency and complexity associated with sepsis diagnosis, models used in this context must not only be accurate but also offer transparency in how decisions are derived.

By applying GWO-FNN to this dataset, we aim to demonstrate its suitability for tasks requiring both predictive performance and interpretability. The results contribute to the

growing body of evidence supporting the use of AI-driven approaches in early sepsis detection, with potential implications for improving clinical outcomes through timely and explainable model-based interventions.

For instance, the application of a novel hybrid metaheuristic algorithm for the optimization of deep neural networks demonstrates a breakthrough in achieving superior accuracy in sepsis diagnosis. This approach combines particle swarm optimization (PSO) and the Human Mental Search algorithm (HMS) to navigate towards the global minimum efficiently, showcasing an innovative direction in AI-driven healthcare solutions [94]. Another pivotal study harnesses unstructured data in healthcare to improve sepsis prediction and diagnosis. It emphasizes the marginal benefits of incorporating clinical text into predictive models, particularly for early prediction windows, thereby highlighting the nuanced complexities in modeling sepsis onset [95].

This real-world application scenario provides a solid foundation to evaluate our model's performance against traditional benchmarks and to showcase its proficiency in unearthing knowledge from data that directly impacts patient care strategies. Table 4 presents some recent research about the area.

Table 4. Summary of recent AI-based sepsis classification studies.

Author	Technique	Achievement
[96]	Clinical Observations	Cognitive Decline Correlation
[97]	Multivariate Analysis	Mortality Prediction
[98]	Prognostic Analysis	New Model Proposal
[99]	Committee-based Machine Learning	Early Prediction of Generalized Infection
[100]	Predictive Analysis	Outcome Predictors in Older Adults
[101]	SVM, Extreme Gradient Boost	Early Sepsis Prediction

These advancements affirm the essence of our research direction—leveraging AI to not only classify sepsis effectively but also to enhance the interpretability and applicability of such models in real-world clinical settings. By focusing on models that excel in sensitivity, specificity, and overall performance, our research aligns with the broader objective of improving sepsis care through early intervention and tailored treatment plans.

Experimental Setup

Our experimental framework is structured to assess the performance and knowledge extraction capabilities of the following models, split in a 70/30 train–test ratio.

- GWO-FNN models: We examine three distinct configurations of the GWO-FNN model—AndNeuron, OrNeuron, and UniNeuron variants—all enhanced with the Adam optimization algorithm and configured with four membership functions for each dimension, following insights from preliminary experiments with varying membership functions (mf = [2, 3, 4,5] and number of wolves = [10, 15, 20]).
- Benchmark models: We employ a suite of well-established models in our comparative analysis, including the following:
    1. RandomForestClassifier with 50 estimators [88];
    2. MLPClassifier with a single hidden layer of 50 neurons and 300 iterations [87];
    3. GaussianNB in its standard configuration [102];
    4. SVC with a linear kernel [103];
    5. LogisticRegression in its default setup [104].

This comparative analysis is designed not only to assess how the proposed models perform relative to established benchmarks but also to explore, in depth, the interpretability

and knowledge representation capabilities of the top-performing GWO-FNN configuration. Understanding the patterns and rules inferred by the model is especially valuable in clinical contexts, where transparent reasoning can support evidence-based decision making and contribute to sepsis research advancements.

In addition, we conduct a comprehensive evaluation of all three GWO-FNN variants, examining their individual impacts on both predictive accuracy and interpretability. The analysis leverages the same performance metrics introduced earlier—accuracy, precision, recall, and F1 score—to maintain consistency across experimental stages and enable direct comparison of the models' strengths and trade-offs.

To facilitate a clear understanding of the dataset dimensions and the distribution of sepsis cases, we present Tables 5 and 6, which delineate the dataset characteristics and sample distribution.

Table 5. Dataset characteristics and sample distribution.

Variable Name	Description	Units/Missing Values
age_years	Age of the patient in years	years/no
sex_0male_1female	Gender of the patient. {0: male, 1: female}	no
episode_number	Number of prior sepsis episodes	no
hospital_outcome_1alive_0dead	Status after 9351 days. {1: alive, 0: dead}	no

Table 6. Feature statistics.

Name	Mean	Median	Dispersion	Min.	Max.	Missing
Age	62.74	68	0.38	0	100	0 (0%)
Episode Number	1.35	1	0.56	1	5	0 (0%)
Hospital Outcome	1	-	0.263	-	-	0 (0%)
Sex	0	-	0.692	-	-	0 (0%)

Through this comprehensive evaluation, we aim to demonstrate our models' profound capability to match and potentially surpass existing methodologies in predicting sepsis outcomes, thereby reinforcing the importance of machine learning models in the critical field of sepsis prognosis.

7. Discussion About the Tests

Upon embarking on this analytical task, it was imperative to meticulously dissect the results procured from previous experimental endeavors. The ensuing discourse aims to illuminate the significance of these findings, not only in the context of validating the hypotheses posited at the outset but also in their capacity to contribute to the broader corpus of knowledge within the field. Through a methodical examination of the data, juxtaposed against the backdrop of established theoretical frameworks, this section endeavors to unveil novel insights and discern patterns that may hitherto have remained obscured. In this way, we intend to furnish a comprehensive understanding of the implications these results hold for both current practices and future investigative pursuits.

7.1. Analysis of Performance Metrics

To assess how different architectural choices influence the GWO-FNN model's behavior in binary classification, we evaluated multiple configurations by varying the type of neuron and the number of membership functions (MFs). This analysis provides valuable insights into the relationship between model structure and predictive performance. Table 7 summarizes key evaluation metrics—accuracy, F1 score, recall, and precision—across the tested configurations, offering a holistic view of each variant's strengths and limitations.

Table 7. Summary of performance metrics across different model configurations.

Neuron Type	MFs	Accuracy	F1 Score	Recall	Precision
AndNeuron	2	0.433	0.605	1.000	0.433
AndNeuron	3	0.667	0.722	1.000	0.565
AndNeuron	4	0.667	0.722	1.000	0.565
OrNeuron	2	**1.000**	**1.000**	**1.000**	**1.000**
OrNeuron	3	**1.000**	**1.000**	**1.000**	**1.000**
OrNeuron	4	**1.000**	**1.000**	**1.000**	**1.000**
UniNeuron	2	0.867	0.867	1.000	0.765
UniNeuron	3	0.667	0.615	0.615	0.615
UniNeuron	4	0.867	0.867	1.000	0.765

An initial performance assessment highlights noticeable differences among the neuron types, especially in scenarios using two and four membership functions, as illustrated in Figure 13. Notably, the configurations employing the OrNeuron and UniNeuron with four membership functions demonstrated exceptional results, achieving maximum scores across all evaluation metrics. These findings emphasize the influence of membership function granularity in enhancing classification effectiveness. The flawless performance observed in these cases suggests that the increased number of MFs enabled a more refined segmentation of the input space, allowing the model to establish highly precise and well-defined decision boundaries.

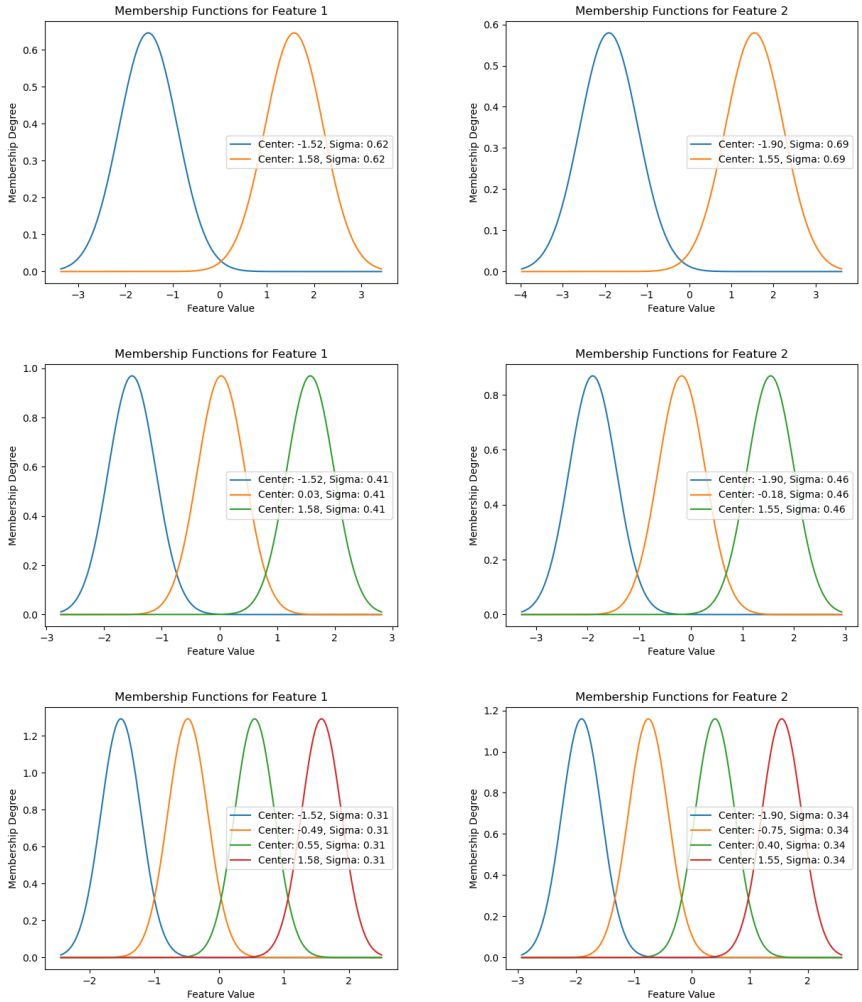


Figure 13. Fuzzification process generated by the model in 2 dimensions.

In contrast, the configuration utilizing the AndNeuron with three membership functions yielded the weakest performance among all tested models, showing notable drops in both accuracy and precision. This suggests that simply increasing the number of MFs within this neuron type may introduce unnecessary complexity rather than improving predictive power. A possible explanation lies in the oversaturation of the input space, which may hinder the model's ability to generalize effectively.

The OrNeuron models consistently achieved superior results, regardless of the number of MFs employed. This stability indicates a robustness in handling varying degrees of input granularity, likely due to the neuron's ability to emphasize relevant features while maintaining strong generalization across binary classification tasks.

UniNeuron-based models exhibited a clear upward trend in performance as the number of MFs increased. This pattern highlights the model's responsiveness to finer feature partitions, reinforcing its potential for fine-tuning and precise classification—particularly in tasks that benefit from higher resolution in fuzzy set representation.

These empirical patterns set the stage for a deeper interpretability analysis, which is further illustrated through visual metrics. Figure 14 presents the similarity levels between the fuzzy rules (with red tones indicating high similarity and blue denoting divergence), while Figure 15 reflects consistency across rule definitions, where more intense red shades represent stronger coherence. Finally, Figure 16 highlights distinguishability, with blue areas signaling potential overlap in rule behavior, pointing to interpretability issues in those regions.

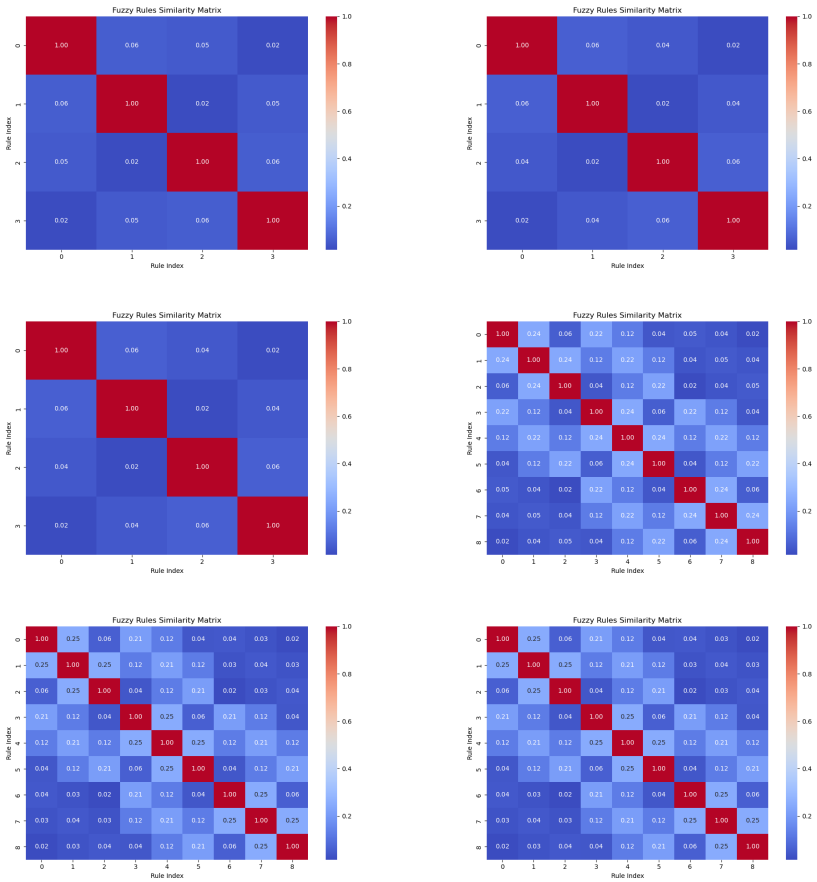


Figure 14. *Cont.*

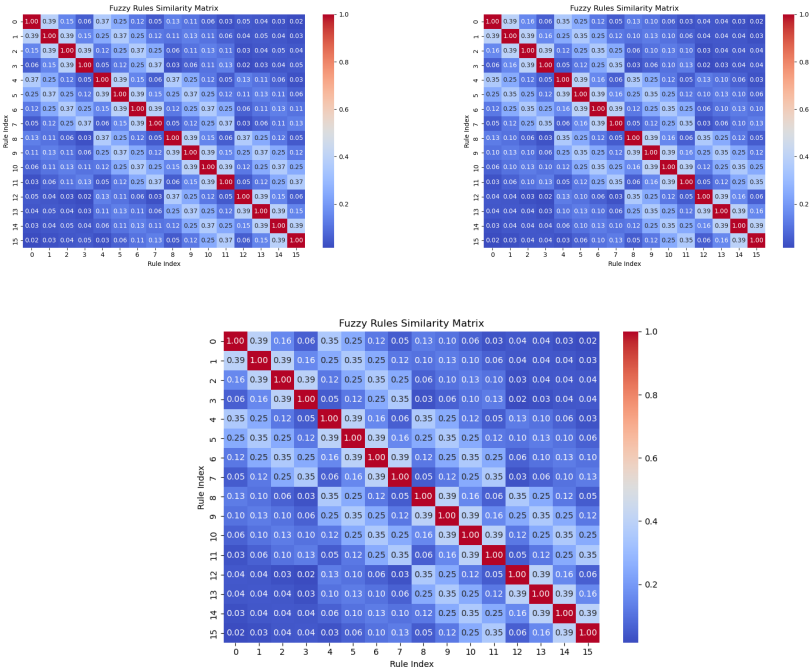


Figure 14. Similarity matrix in the experiments (each column represents AndNeuron, OrNeuron, and UniNeuron for 2 MFs, 3 MFs, and 4 MFs, respectively).

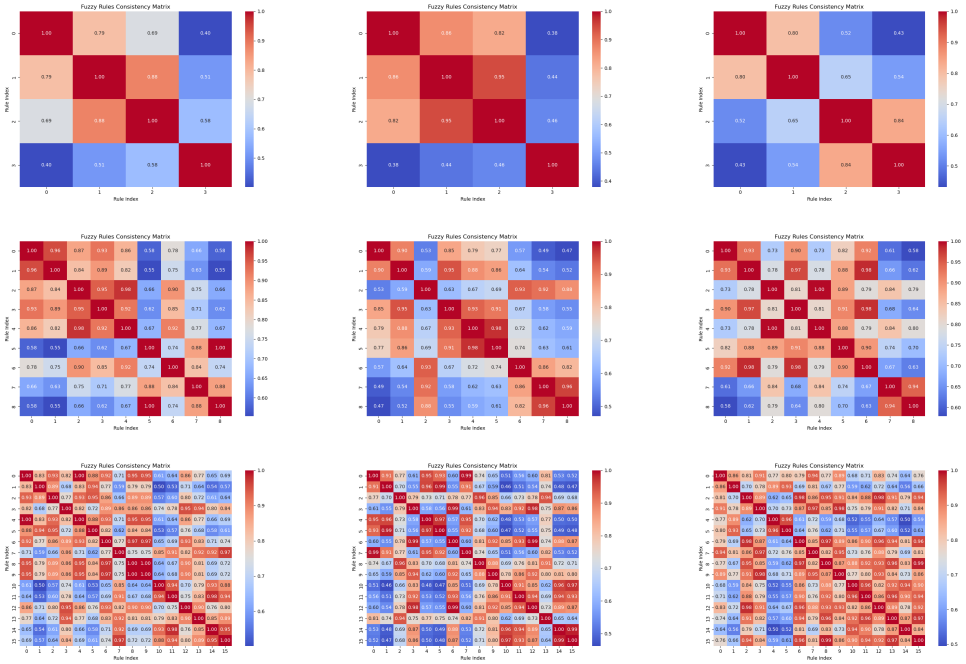


Figure 15. Consistency matrix in the experiments (each column represents AndNeuron, OrNeuron, and UniNeuron for 2 MFs, 3 MFs, and 4 MFs, respectively).

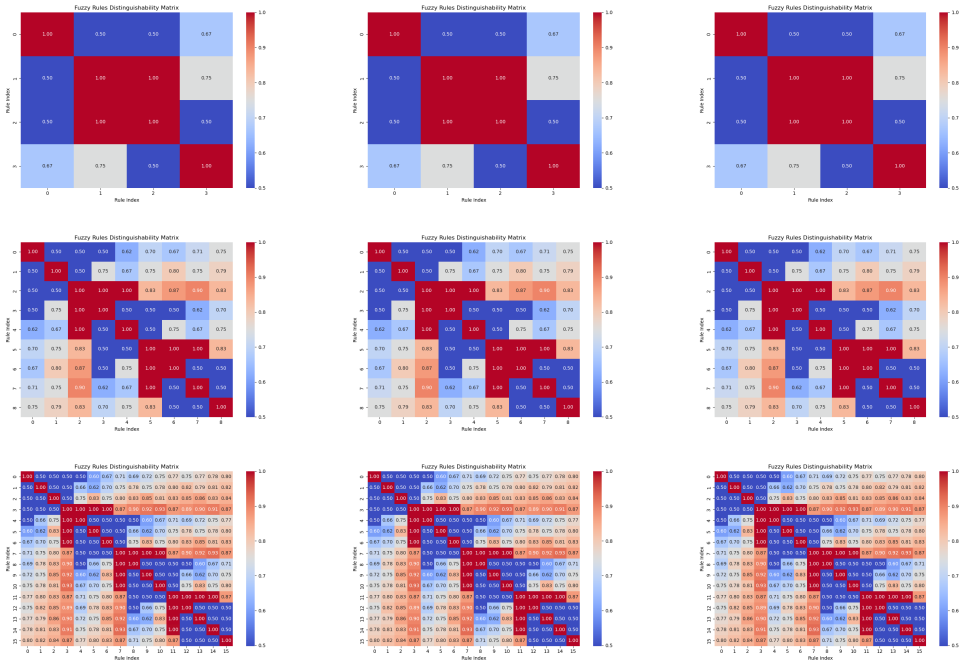


Figure 16. Distinguishability matrix in the experiments: AndNeuron, OrNeuron, and UniNeuron for 2 MFs, 3 MFs, and 4 MFs, respectively.

Finally, the ϵ-completeness criterion (Figure 17) indicates whether the membership functions adequately covered the samples during the training phase. It was observed that the AndNeuron model encountered significant issues with coverage throughout the experiment, highlighting its diminished performance. This aspect presents a compelling avenue for exploration in subsequent research endeavors. Understanding the interplay between model complexity, interpretability, and performance is essential, guiding us towards configurations that not only achieve high accuracy but also maintain clarity and interpretability in decision-making processes.

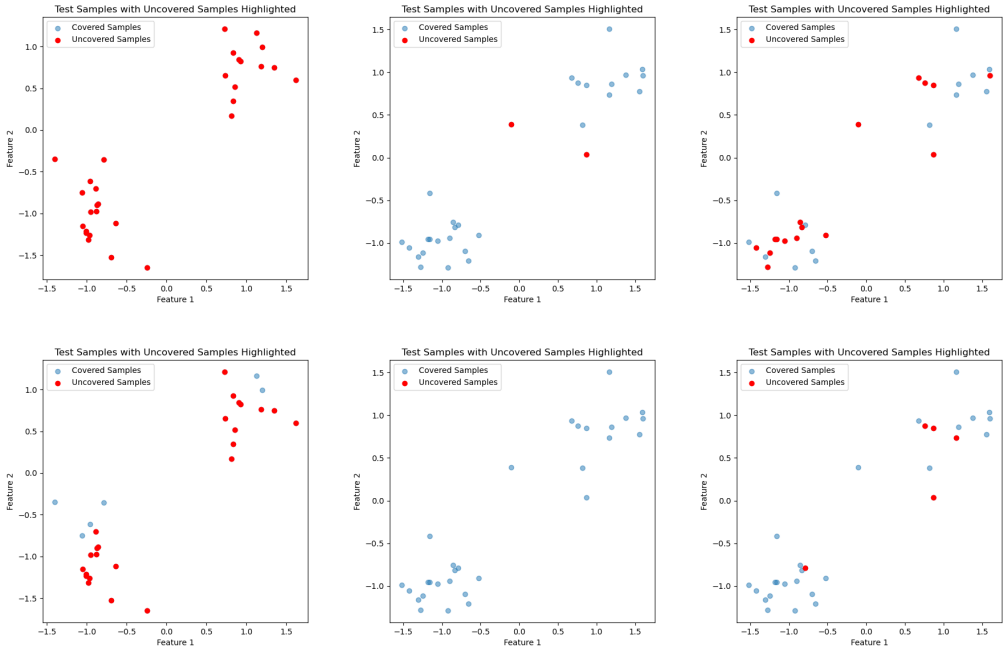


Figure 17. *Cont.*



Figure 17. ϵ-completeness plot in the experiments (each column represents AndNeuron, OrNeuron, and UniNeuron for 2 MFs, 3 MFs, and 4 MFs, respectively).

7.2. State-of-the-Art Evaluation

This subsection articulates the outcomes of experiments that have been previously delineated. The findings from the thirty iterations conducted within the experiment are cataloged in Table 8, wherein the mean of the evaluated scores, along with the standard deviation encapsulated in parentheses, are presented.

In analyzing the performance of FNNs across different datasets (Table 8), it is evident that the effectiveness of these models in classification tasks is contingent upon the characteristics of the data they are applied to. The metrics of accuracy, precision, recall, and F1 score serve as pivotal indicators of performance, guiding the optimal deployment of GWO-FNN configurations for specific scenarios.

The GWO-FNN models exhibit exemplary performance on the Iris dataset, achieving perfect scores (1.00) across all metrics. This outcome highlights the models' proficiency in handling datasets with distinct, linearly separable classes, where the relationships between features and classes are direct and well defined.

Contrastingly, the performance on the Mammographic Masses dataset delineates the variability inherent in GWO-FNN applications. The AndNeuron variant of GWO-FNN shows reduced performance, particularly in accurately classifying complex, real-world medical imaging data that often contain intricate patterns and noise. On the other hand, the OrNeuron configuration of GWO-FNN demonstrates enhanced capability, achieving higher precision and recall values. This distinction underscores the robustness of certain GWO-FNN configurations in managing the precision–recall balance essential for effectively dealing with the nuanced classification challenges presented by such datasets.

The examination of GWO-FNN models across the Transfusion, Immunotherapy, and Cryotherapy datasets further reveals their flexible applicability. These medical datasets, each with unique classification challenges, test the generalizability and efficacy of GWO-FNN models, underscoring the critical role of model selection and tuning in achieving optimal performance.

Notably, the GWO-FNN OrNeuron model consistently emerges as particularly effective, especially in datasets characterized by complex classification landscapes like those found in medical imaging. This model's superior performance is indicative of the importance of carefully selecting and adjusting GWO-FNN architectures and neuron configurations to suit the specific demands and characteristics of each dataset.

The analysis collectively suggests that while GWO-FNN models hold significant promise for classification tasks, their success is highly dependent on the alignment of model configurations with dataset specifics. This highlights the nuanced nature of applying FNNs, where strategic considerations regarding architecture and neuron type are crucial for navigating the challenges of diverse classification contexts and achieving the best possible outcomes.

Table 8. The performance of FNNs across different datasets.

Model	Accuracy	Precision	Recall	F1 Score
Dataset: Iris				
GWO-FNN AndNeuron	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
GWO-FNN OrNeuron	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
GWO-FNN UniNeuron	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
Random Forest	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
MLP	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
Naive Bayes	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
Dataset: Mammographic Masses				
GWO-FNN AndNeuron	0.54 (0.01)	1.00 (0.00)	0.02 (0.01)	0.04 (0.03)
GWO-FNN OrNeuron	0.83 (0.01)	0.76 (0.02)	0.92 (0.00)	0.83 (0.01)
GWO-FNN UniNeuron	0.74 (0.02)	0.73 (0.03)	0.70 (0.02)	0.72 (0.02)
Random Forest	0.84 (0.00)	0.83 (0.00)	0.83 (0.00)	0.83 (0.00)
MLP	0.88 (0.00)	0.87 (0.00)	0.87 (0.00)	0.87 (0.00)
Naive Bayes	0.84 (0.00)	0.79 (0.00)	0.91 (0.00)	0.84 (0.00)
Dataset: Haberman				
GWO-FNN AndNeuron	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
GWO-FNN OrNeuron	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
GWO-FNN UniNeuron	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
Random Forest	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
MLP	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
Naive Bayes	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
Dataset: Transfusion				
GWO-FNN AndNeuron	0.78 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
GWO-FNN OrNeuron	0.77 (0.00)	0.47 (0.01)	0.25 (0.03)	0.32 (0.02)
GWO-FNN UniNeuron	0.76 (0.01)	0.33 (0.01)	0.08 (0.04)	0.12 (0.05)
Random Forest	0.68 (0.00)	0.29 (0.01)	0.28 (0.00)	0.28 (0.00)
MLP	0.78 (0.00)	0.51 (0.01)	0.31 (0.01)	0.39 (0.01)
Naive Bayes	0.78 (0.00)	0.50 (0.00)	0.16 (0.00)	0.24 (0.00)
Dataset: Liver Data				
GWO-FNN AndNeuron	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
GWO-FNN OrNeuron	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
GWO-FNN UniNeuron	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
Random Forest	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
MLP	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
Naive Bayes	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
Dataset: Immunotherapy				
GWO-FNN AndNeuron	0.78 (0.00)	0.78 (0.00)	1.00 (0.00)	0.88 (0.00)
GWO-FNN OrNeuron	0.74 (0.07)	0.82 (0.02)	0.86 (0.10)	0.84 (0.05)
GWO-FNN UniNeuron	0.80 (0.02)	0.79 (0.02)	1.00 (0.00)	0.88 (0.01)
Random Forest	0.80 (0.02)	0.82 (0.02)	0.95 (0.00)	0.88 (0.01)
MLP	0.72 (0.02)	0.80 (0.02)	0.86 (0.00)	0.83 (0.01)
Naive Bayes	0.59 (0.00)	0.75 (0.00)	0.71 (0.00)	0.73 (0.00)
Dataset: Cryotherapy				
GWO-FNN AndNeuron	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
GWO-FNN OrNeuron	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
GWO-FNN UniNeuron	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
Random Forest	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
MLP	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
Naive Bayes	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)

For a safe validation of the results, a statistical analysis (variance analysis) was carried out to verify whether the performance of the models proposed in this paper is similar to the state of the art in the classification of binary patterns. The results are presented in Table 9.

Table 9. ANOVA results.

Source	sum_sq	df	F	PR (>F)
C(Model)	0.083	5	78.682	2.038×10^{-56}
C(Dataset)	6.046	6	4797.360	0.000×10^{00}
C(Model):C(Dataset)	1.055	30	167.438	2.708×10^{-198}
Residual	0.079	378	NaN	NaN

The ANOVA test validation assumptions are listed in Table 10.

Table 10. Shapiro–Wilk and Levene's test results.

Test	Statistics	*p*-Value
Shapiro–Wilk	0.368	1.16×10^{-35}
Levene's	0.953	0.447

In Table 11, the models considered statistically equivalent by the Tukey test (with a high p-Tukey value, indicating that the difference between the means is not statistically significant) are highlighted in gray. This allows you to quickly view comparisons between models where there was no significant difference, making it easier to identify models with equivalent performance.

The statistical evaluation of fuzzy neural network (FNN) models across various datasets, as reflected by ANOVA and Tukey's honest significant difference (HSD) test results, provides a nuanced understanding of the models' performance and their interactions with specific data characteristics. The ANOVA outcomes highlight significant variances attributable to both the choice of model and dataset, underscoring the different impacts these factors have on classification efficacy. Notably, the interaction term between model and dataset further suggests that the performance of any given GWO-FNN model is intricately tied to the dataset it is applied to, indicating that no single model uniformly outperforms others across all contexts.

Table 11. Tukey's HSD test results.

A	B	Mean (A)	Mean (B)	Diff	se	T	p-Tukey	Hedges
FNN AndNeuron	FNN OrNeuron	0.870	0.913	−0.043	0.022	−1.930	0.385	−0.305
FNN AndNeuron	FNN UniNeuron	0.870	0.903	−0.033	0.022	−1.505	0.661	−0.231
FNN AndNeuron	MLP	0.870	0.902	−0.031	0.022	−1.424	0.713	−0.212
FNN AndNeuron	Naive Bayes	0.870	0.888	−0.018	0.022	−0.796	0.968	−0.111
FNN AndNeuron	Random Forest	0.870	0.904	−0.034	0.022	−1.520	0.651	−0.230
FNN OrNeuron	FNN UniNeuron	0.913	0.903	0.010	0.022	0.425	0.998	0.087
FNN OrNeuron	MLP	0.913	0.902	0.011	0.022	0.507	0.996	0.098
FNN OrNeuron	Naive Bayes	0.913	0.888	0.025	0.022	1.134	0.867	0.197
FNN OrNeuron	Random Forest	0.913	0.904	0.009	0.022	0.410	0.999	0.081
FNN UniNeuron	MLP	0.903	0.902	0.001	0.022	0.081	1.000	0.015
FNN UniNeuron	Naive Bayes	0.903	0.888	0.015	0.022	0.709	0.981	0.119
FNN UniNeuron	Random Forest	0.903	0.904	−0.001	0.022	−0.015	1.000	−0.003
MLP	Naive Bayes	0.902	0.888	0.014	0.022	0.628	0.989	0.101
MLP	Random Forest	0.902	0.904	−0.002	0.022	−0.096	1.000	−0.017
Naive Bayes	Random Forest	0.888	0.904	−0.016	0.022	−0.724	0.979	−0.119

The Shapiro–Wilk and Levene's tests, assessing the normality of residuals and equality of variances, respectively, provide information on the underlying assumptions of the ANOVA analysis. While the Shapiro–Wilk test indicates a departure from normality, the

Levene's test confirms homogeneity of variances, ensuring that the subsequent analyses, including Tukey's HSD, are based on sound statistical footing.

Tukey's HSD test, a post hoc analysis, further dissects the ANOVA results to identify specific pairs of models where performance differences are statistically significant or negligible. This pairwise comparison is critical in distinguishing between models that are statistically equivalent in their performance and those that are not. For instance, the statistical equivalence between the GWO-FNN OrNeuron and GWO-FNN UniNeuron models across certain datasets suggests that their classification performances are indistinguishable, offering flexibility in model selection based on other criteria such as computational efficiency or ease of implementation.

Conversely, significant differences in performance metrics between other model pairs highlight the importance of careful model selection tailored to the specific characteristics and requirements of the dataset in question. Such distinctions underscore the potential for optimizing classification outcomes by aligning model capabilities with dataset nuances.

In essence, the statistical analysis reveals the complexity of model performance across datasets, emphasizing that the effectiveness of the GWO-FNN models is context-dependent. This underscores the necessity for a deliberate and informed approach to model selection, one that considers both statistical metrics and practical considerations, to harness the full potential of GWO-FNNs in varied classification tasks.

7.3. Analysis of the Sepsis Identification Results

This subsection presents a detailed performance assessment of our proposed models applied to the sepsis dataset. We compare the outcomes of three Adam-optimized fuzzy neural network (FNN) variants—AndNeuron, OrNeuron, and UniNeuron—against several well-established machine learning algorithms. These include Random Forest, Multilayer Perceptron (MLP), Naive Bayes, Support Vector Machine (SVM), and Logistic Regression.

To ensure a fair comparison, all models were evaluated using four key classification metrics: accuracy, precision, recall, and F1 score. The results, summarized in Table 12, offer a comprehensive view of how each model performed in the context of sepsis detection, highlighting their respective strengths and limitations in terms of both predictive quality and generalization ability.

Table 12. Performance metrics of models on the sepsis dataset.

Model	Accuracy	Precision	Recall	F1 Score
FNN AndNeuron	0.927	0.962	1.000	0.927
FNN OrNeuron	0.927	0.927	1.000	0.962
FNN UniNeuron	0.927	0.926	1.000	0.962
Random Forest	0.926	0.926	1.000	0.962
MLP	0.926	0.926	1.000	0.962
Naive Bayes	0.926	0.926	1.000	0.962
SVM	0.926	0.926	1.000	0.962
Logistic Regression	0.926	0.926	1.000	0.962

Analysis

The results underscore the Adam-optimized FNN models' impressive performance, demonstrating their capability to match the state-of-the-art models in terms of accuracy, precision, recall, and F1 score on the sepsis dataset. Notably, the FNN variants exhibit perfect recall, highlighting their efficiency in identifying all relevant sepsis cases without any false negatives. This equivalence to traditional machine learning models, especially in achieving a perfect recall and comparable precision, solidifies the Adam-optimized FNN models as a compelling choice for clinical applications where identifying every possible

case of sepsis is crucial. The uniform performance across different metrics suggests that these models can be reliably used in the medical field, offering an excellent balance between detecting true positives and minimizing false positives.

In the conducted experiment, Figure 18 delineates the outcomes of the fuzzification process, clearly demarcating the formation of result groups. For enhanced interpretability, the evaluation incorporates two significant functions of relevance, achieving outcomes that are statistically on par with leading-edge models. This approach facilitates the interpretation of numerical dimensions, where the inaugural membership function corresponds to lesser magnitudes within the dimension. For instance, in the context of age, distinctions are made between high age and advanced age, thereby bridging human a priori knowledge with the insights gleaned from the model's analysis. The fuzzy rules generated throughout the experiment are enumerated in Tables 13–15. Significantly, the model introduced in this study achieves comparable results utilizing a more streamlined architectural framework, comprising merely nine fuzzy rules.

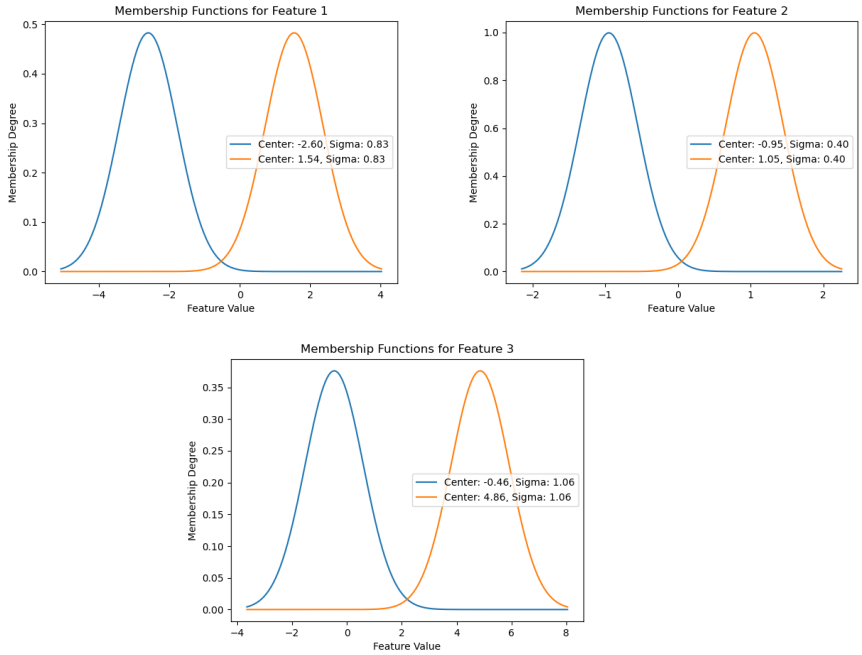


Figure 18. Fuzzification process generated by the model in 2 dimensions on sepsis dataset.

Note: The use of different shades of color to emphasize the certainty of a consequent is a valued feature of this model, warranting a detailed explanation. As noted in Table 13, the numbers in parentheses next to each membership function (MF) represent the impact of that MF on the outcome. The last column uses blue and red colors to indicate whether the patient survived or did not, respectively. This gradation of colors reflects the relationship of the consequents established by the Adam optimizer: weights vary between -1 and 1. When the rule's consequent is very close to -1, it is depicted with a stronger shade of red, indicating a high probability of the class -1 occurrence (non-survivor). Similarly, a stronger shade of blue signifies that the consequent of the rule approaches 1, suggesting a strong likelihood of survival. When the consequent is near zero, the rule does not significantly contribute to swaying the decision towards any class, which can also be seen as an ambiguity in the decision-making process. This visual approach not only facilitates the interpretation of the results but also highlights the model's ability to quantify the certainty associated with each prediction.

Table 13. Fuzzy rules generated by AndNeuron. Note: Parentheses indicate the impact on the membership function (MF) generated by the weights using mutual information.

Rule	Conseq.	Situation
IF Age is MF1 (0.51) AND Sex is MF1 (0.41) AND Ep. number is MF1 (0.37)	12.84	Alive
IF Age is MF1 (0.43) AND Sex is MF1 (0.21) AND Ep. number is MF2 (0.09)	1.98	Alive
IF Age is MF1 (0.58) AND Sex is MF2 (0.77) AND Ep. number is MF1 (0.45)	15.42	Alive
IF Age is MF1 (0.39) AND Sex is MF2 (0.63) AND Ep. number is MF2 (0.33)	3.07	Alive
IF Age is MF2 (0.19) AND Sex is MF1 (0.62) AND Ep. number is MF1 (0.55)	−4.11	Deceased
IF Age is MF2 (0.48) AND Sex is MF1 (0.43) AND Ep. number is MF2 (0.86)	0.71	Alive
IF Age is MF2 (0.12) AND Sex is MF2 (0.91) AND Ep. number is MF1 (0.88)	−3.24	Deceased
IF Age is MF2 (0.34) AND Sex is MF2 (0.68) AND Ep. number is MF2 (0.47)	1.37	Alive

Table 14. Fuzzy rules generated by OrNeuron. Note: Parentheses indicate the impact on the membership function (MF) generated by the weights using mutual information.

Rule	Conseq.	Situation
IF Age is MF1 (0.33) OR Sex is MF1 (0.26) OR Ep. number is MF1 (0.18)	1.17	Alive
IF Age is MF1 (0.15) OR Sex is MF1 (0.64) OR Ep. number is MF2 (0.85)	1.61	Alive
IF Age is MF1 (0.23) OR Sex is MF2 (0.50) OR Ep. number is MF1 (0.77)	1.45	Alive
IF Age is MF1 (0.11) OR Sex is MF2 (0.90) OR Ep. number is MF2 (0.61)	4.98	Alive
IF Age is MF2 (0.44) OR Sex is MF1 (0.69) OR Ep. number is MF1 (0.72)	0.38	Alive
IF Age is MF2 (0.28) OR Sex is MF1 (0.52) OR Ep. number is MF2 (0.33)	−0.81	Deceased
IF Age is MF2 (0.76) OR Sex is MF2 (0.38) OR Ep. number is MF1 (0.64)	−0.41	Deceased
IF Age is MF2 (0.41) OR Sex is MF2 (0.51) OR Ep. number is MF2 (0.13)	−1.21	Deceased

Table 15. Fuzzy rules generated by UniNeuron. Note: Parentheses indicate the impact on the membership function (MF) generated by the weights using mutual information.

Rule	Conseq.	Situation
IF Age is MF1 (0.59) OR Sex is MF1 (0.39) OR Ep. number is MF1 (0.21)	0.41	Alive
IF Age is MF1 (0.37) AND Sex is MF1 (0.76) AND Ep. number is MF2 (0.84)	−0.11	Deceased
IF Age is MF1 (0.16) AND Sex is MF2 (0.48) AND Ep. number is MF1 (0.87)	0.19	Alive
IF Age is MF1 (0.68) AND Sex is MF2 (0.79) AND Ep. number is MF2 (0.50)	0.63	Alive
IF Age is MF2 (0.62) OR Sex is MF1 (0.29) OR Ep. number is MF1 (0.47)	−0.07	Imprecise
IF Age is MF2 (0.66) AND Sex is MF1 (0.37) AND Ep. number is MF2 (0.38)	−0.03	Imprecise
IF Age is MF2 (0.36) OR Sex is MF2 (0.74) OR Ep. number is MF1 (0.91)	−0.41	Deceased
IF Age is MF2 (0.11) AND Sex is MF2 (0.49) AND Ep. number is MF2 (0.73)	0.05	Imprecise

In conducting a thorough analysis of the fuzzy inference rules generated by different models for a medical journal, focusing on "Age", "Sex", and "Number of Episodes" as pivotal features, we uncover distinct patterns, similarities, and differences. This analytical lens allows us to decipher what each model reveals about the complex interplay of these factors in determining sepsis outcomes. The following presents the patterns revealed across the models.

- Impact of age: All models underscore the significant role of age in predicting sepsis outcomes. Rules with "Age" as a contributing factor often suggest that older age groups are associated with higher risks, reflecting a widely recognized clinical observation. The mathematical rigor behind this is apparent in the rules, where age's impact scores directly correlate with the predicted outcomes, offering a quantitative basis for this age-related vulnerability.
- Influence of sex: While less pronounced than age, the variable "Sex" also features prominently across the models. The differentiation based on sex, albeit with varying degrees of impact across the models, hints at physiological or possibly social determinants influencing sepsis survival rates. This inclusion of sex as a variable enriches the models' capacity to tailor predictions more closely to individual patient profiles.
- Number of episodes as a critical marker: Perhaps the most telling is the emphasis on the "Number of Episodes". Models tend to associate a higher number of sepsis episodes with increased mortality risk, but the degree of impact and the manner in which this variable

interacts with others (age and sex) vary. This variability offers nuanced insights into how recurrent sepsis episodes compound risk, a crucial consideration for clinicians.

Here are some similarities and differences between the fuzzy rules generated.

- Similarities:
  1. Consistent representation of risk factors: Despite the inherent differences in their formulation, the rules across models maintain a consistent representation of "Age" and "Number of Episodes" as critical risk factors. This consistency offers a unified view of these variables' importance, facilitating a broader understanding of their impact on sepsis outcomes.
  2. Logical structuring of antecedents and consequents: The models adhere to a logical structuring that maps specific combinations of antecedents (variables and their states) to consequents (predicted outcomes). This structuring aids in the interpretability of the rules by clearly delineating the conditions under which certain outcomes are expected, thereby providing insights into the decision-making process of the models.
  3. Quantitative insights through impact scores: All models utilize impact scores to quantify the influence of each antecedent on the consequent, offering a measurable insight into the significance of each risk factor. This approach not only enhances the interpretability of the rules but also provides a basis for comparing the relative importance of different variables within the same rule, further enriching the models' analytical depth.
- Interpretability through dimensional focus: Across all models, there is a clear emphasis on making the rules interpretable by focusing on clinically relevant dimensions such as "Age" and "Number of Episodes". This focus not only aligns with clinical priorities but also enhances the models' usability by providing clear, actionable insights into how key variables influence sepsis prognosis.
- Differences:
  1. Conflicting outcomes from similar antecedents: An analysis reveals instances where identical or similar antecedents across different rules lead to divergent outcomes. For example, one rule might indicate a high risk of mortality with a specific configuration of "Age" and "Number of Episodes", whereas another rule with the same antecedents suggests a survival outcome. This discrepancy can stem from the subtle nuances in how each model weighs the significance or impact of these features differently, reflecting the inherent complexity and variability in sepsis prognosis.
  2. Variable interactions: The rules also differ in their portrayal of interactions between variables. In some models, "Age" and "Number of Episodes" might interact synergistically, amplifying each other's effects. In contrast, other models might present these interactions as more independent, with each variable contributing to the outcome in a more isolated manner. This variation underscores different models' interpretations of how sepsis risk factors interrelate.
  3. Impact score variability: Even within models that use the same logical structure (AND/OR), the assigned impact scores for similar antecedents can vary significantly, leading to different conclusions. This aspect highlights the models' flexibility but also introduces challenges in consistently interpreting the influence of specific risk factors on sepsis outcomes.

These rules constitute a robust knowledge base, adept at addressing sepsis challenges with an accuracy exceeding 92%. The subsequent figures showcase evaluations of similarity (Figure 19), consistency (Figure 20), and ϵ-completeness (Figure 21) across the fuzzy rules for each model version examined. Through the consistency matrix, it is feasible to pinpoint rules that might provoke conflicts by identifying those tinted bluer (denoting lesser

consistency). In this instance, it is observed that for the OrNeuron, the most inconsistent rule is rule 4. This rule could become a focal point for forthcoming analytical endeavors or systemic interventions, aiming to remove it and assess the impact on the results.

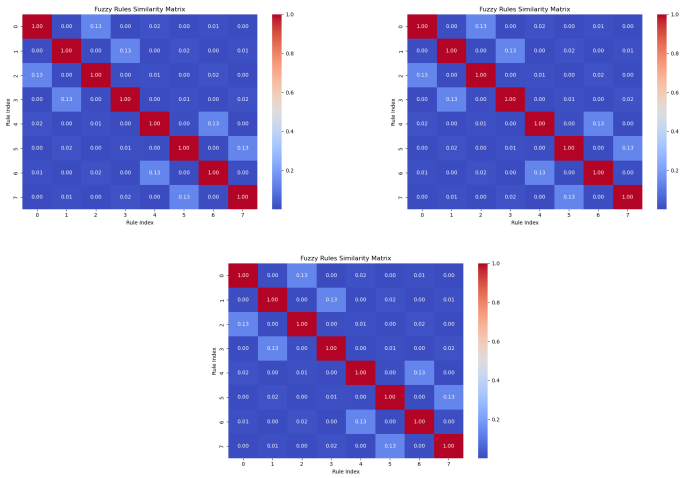


Figure 19. Consistency of fuzzy rules on sepsis dataset: AndNeuron, OrNeuron, and UniNeuron.

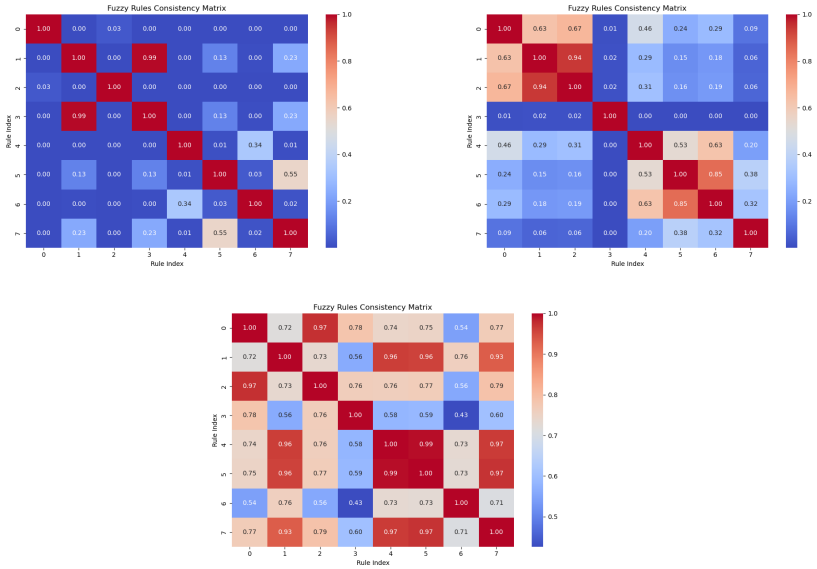


Figure 20. Similarities matrices of fuzzy rules on sepsis dataset.

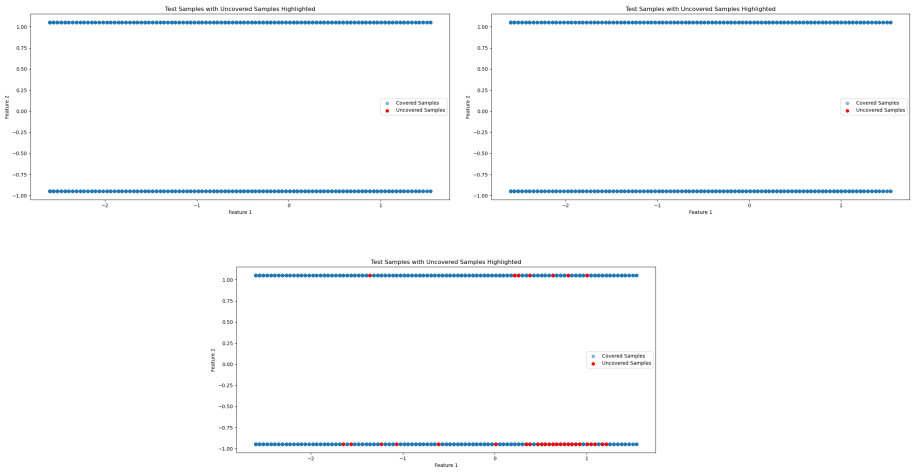


Figure 21. Similarities of fuzzy rules on sepsis dataset.

8. Conclusions

The GWO-FNN model presents an integrated approach that balances interpretability and predictive performance by leveraging fuzzy logic and neural networks. Utilizing the Grey Wolf Optimizer (GWO) for optimization, the model achieves enhanced accuracy while maintaining transparency in decision making. This advancement contributes to the ongoing development of AI systems that prioritize explainability without sacrificing computational efficiency.

8.1. Key Contributions

This study highlights the effectiveness of the proposed model in extracting meaningful data insights through rigorous statistical validation and interpretability assessments. A central innovation lies in the transformation of logical neurons into explicit fuzzy rules, ensuring that the model's reasoning process remains transparent. The structured analysis of rule importance, sensitivity, and completeness further supports the development of interpretable AI models, offering clear justifications for predictions.

The model's application to a sepsis case study demonstrates its ability to provide state-of-the-art predictive performance while simplifying rule-based decision making. This capability is particularly valuable in medical and high-stakes domains, where interpretability is crucial for informed decision making. Beyond the medical domain, the GWO-FNN framework can be applied to other critical areas such as financial risk assessment, industrial quality control, and cybersecurity, where explainability is a fundamental requirement. Moreover, the methodology can be extended to other applications requiring a balance between accuracy and explainability.

8.2. Challenges and Future Work

Despite its strengths, the model encounters computational challenges when applied to large datasets with high-dimensional features. The reliance on equally spaced membership functions may introduce inefficiencies, impacting scalability. Additionally, handling complex, high-dimensional spaces remains a limitation of the current framework.

Future research will focus on improving the model's ability to manage large-scale data and high-dimensional structures. One promising direction is the exploration of adaptive or data-driven membership function placement to enhance flexibility and reduce computational overhead. Potential solutions include the implementation of pruning techniques to reduce computational complexity and enhance model efficiency. Further advancements in interpretability methodologies will also be explored to reinforce the model's applicability in real-world decision-making scenarios, including finance, industrial automation, and cybersecurity.

In summary, the GWO-FNN model advances the integration of interpretability and performance in AI systems. While it effectively enhances explainability in predictive modeling, addressing scalability challenges will be crucial for its broader adoption and impact. Expanding its validation across diverse real-world datasets and incorporating hybrid optimization techniques could further solidify its effectiveness in practical applications. Future developments should prioritize both computational efficiency and interpretability to further the progress of transparent and reliable AI models.

Author Contributions: Conceptualization, P.V.d.C.S. and I.S.; methodology, P.V.d.C.S.; validation, P.V.d.C.S.; formal analysis, P.V.d.C.S.; investigation, P.V.d.C.S.; resources, P.V.d.C.S.; data curation, P.V.d.C.S.; writing—original draft preparation, P.V.d.C.S.; writing—review and editing, P.V.d.C.S. and I.S.; visualization, I.S.; supervision, P.V.d.C.S.; project administration, P.V.d.C.S.; funding acquisition, P.V.d.C.S. All authors have read and agreed to the published version of the manuscript.

Funding: We acknowledge the support of the PNRR project FAIR-Future AI Research (PE00000013), under the NRRP MUR program funded by the NextGenerationEU.

Data Availability Statement: All datasets used in this study are public and freely accessible for research purposes. The website used for data collection is the UCI Machine Learning Repository https://archive.ics.uci.edu/ (accessed on 27 March 2025).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ALMMo	Autonomous learning multi-model
ANFIS	Adaptive neuro-fuzzy inference system
ALS	Autonomous learning system

Appendix A

Table A1. Nomenclature and notation used in this study.

Symbol	Description
x, y	Input elements
$\hat{y}$	Predicted class label
A, B	General fuzzy set values
A_{lj}	Fuzzy set for input variable x_j at partition l
$\mu_A(x)$	Membership function of fuzzy set A
$\mu_{B_j}(x)$	Activation degree of fuzzy rule B_j given input x
c	Center of Gaussian membership function
σ	Spread (standard deviation) of Gaussian membership function
$\omega(x_j, c_{jl}, \sigma_{jl})$	Gaussian membership function
a_{jl}	Membership degree of input x_j in fuzzy set A_{lj}
c_{jl}	Center of the Gaussian neuron for fuzzy set A_{lj}
σ_{jl}	Standard deviation of the Gaussian neuron for fuzzy set A_{lj}
$I(x_j, y)$	Mutual information (MI) between feature x_j and class label y
$P(x_j, y)$	Joint probability distribution of feature x_j and label y
$P(x_j), P(y)$	Marginal probabilities of feature x_j and label y
w_j	Initial weight based on mutual information
w_{min}, w_{max}	Predefined range for MI-based weight normalization
$\text{AND}(x, y)$	Boolean AND operation (product)
$\text{OR}(x, y)$	Boolean OR operation (probabilistic sum)
$T(A, B)$	Fuzzy AND operation (t-norm)
$S(A, B)$	Fuzzy OR operation (t-conorm)
$t(x, y)$	T-norm operation (product)
$s(x, y)$	T-conorm operation (probabilistic sum)
$U(x, y, g)$	Uninorm function
g	Identity element in the unit interval $[0, 1]$
BO	Binary operator in $[0, 1]^2 \to [0, 1]$
$\mathbf{a}$	Vector of fuzzy relevance values: $[a_1, a_2, \ldots, a_N]$
$\mathbf{w}$	Vector of weights: $[w_1, w_2, \ldots, w_N]$
$\mathbf{z}$	Output of the logical neuron
w_j	Weight associated with fuzzy rule B_j
m	Total number of fuzzy rules
w_{il}	Weight assigned to feature i in fuzzy neuron l
v_k	Optimized weight of rule consequent k
L	Number of fuzzy neurons per feature

Table A1. *Cont.*

Symbol	Description
$\vec{X}$	Position vector of a gray wolf
$\vec{X}_p$	Position of the prey
$\vec{X}_\alpha, \vec{X}_\beta, \vec{X}_\delta$	Positions of the alpha, beta, and delta wolves
$\vec{D}$	Distance between wolf and prey
$\vec{D}_\alpha, \vec{D}_\beta, \vec{D}_\delta$	Distances from wolves to the best solutions
$\vec{A}, \vec{C}$	Coefficient vectors in the GWO
$\vec{r}_1, \vec{r}_2$	Random vectors in [0, 1]
$\vec{a}$	Linearly decreasing parameter (from 2 to 0)
t	Current iteration number
T	Maximum number of iterations
$\vec{v}_k^{(t)}$	Weight vector of neuron k at iteration t
$\vec{v}_\alpha^{(t)}$	Best solution found so far (alpha wolf position) at iteration t
A	Adaptive control parameter in GWO
D	Distance between candidate solution and leader in GWO
C	Number of classes in classification task
$\vec{Z}$	Activation outputs of all neurons for input samples
$\vec{y}_k$	Target vector for binary classification
Fitness$(\vec{v})$	Fitness function evaluating the quality of $\vec{v}_k$
y_i	Actual class label for sample i
$f_\Gamma(z_i)$	Network's output after applying Sigmoid activation
$f_\Gamma(z)$	Sigmoid activation function
e^{-z}	Exponential term in the Sigmoid function
Ω	Threshold-based classification function
z_0	Bias term (fixed at 1)
v_0	Bias weight
v_j	Weight in the aggregation step
l	Number of aggregation neuron inputs
z_j	Output of neuron j before applying the Sigmoid function
$Rule_L$	Fuzzy rule notation
$\uplus_L$	Output of the fuzzy rule consequent for rule L
$v_1, v_2, \ldots, v_L$	Singleton values for binary classification

References

1. Guidotti, R.; Monreale, A.; Ruggieri, S.; Turini, F.; Giannotti, F.; Pedreschi, D. A survey of methods for explaining black box models. *ACM Comput. Surv. (CSUR)* **2018**, *51*, 1–42. [CrossRef]
2. de Campos Souza, P.V. Fuzzy neural networks and neuro-fuzzy networks: A review the main techniques and applications used in the literature. *Appl. Soft Comput.* **2020**, *92*, 106275. [CrossRef]
3. Angelov, P.P.; Zhou, X. Evolving fuzzy-rule-based classifiers from data streams. *IEEE Trans. Fuzzy Syst.* **2008**, *16*, 1462–1475. [CrossRef]
4. Škrjanc, I.; Iglesias, J.A.; Sanchis, A.; Leite, D.; Lughofer, E.; Gomide, F. Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: A survey. *Inf. Sci.* **2019**, *490*, 344–368. [CrossRef]
5. Pedrycz, W. Fuzzy neural networks and neurocomputations. *Fuzzy Sets Syst.* **1993**, *56*, 1–28. [CrossRef]
6. Wu, S.; Er, M.J.; Gao, Y. A fast approach for automatic generation of fuzzy rules by generalized dynamic fuzzy neural networks. *IEEE Trans. Fuzzy Syst.* **2001**, *9*, 578–594.
7. Slawinski, T.; Krone, A.; Hammel, U.; Wiesmann, D.; Krause, P. A hybrid evolutionary search concept for data-based generation of relevant fuzzy rules in high dimensional spaces. In Proceedings of the FUZZ-IEEE'99. 1999 IEEE International Fuzzy Systems. Conference Proceedings (Cat. No.99CH36315), Seoul, Republic of Korea, 22–25 August 1999; Volume 3, pp. 1432–1437. [CrossRef]
8. Mendel, J.M.; John, R.B. Type-2 fuzzy sets made simple. *IEEE Trans. Fuzzy Syst.* **2002**, *10*, 117–127. [CrossRef]
9. Jang, J.S.; Sun, C.T. Neuro-fuzzy modeling and control. *Proc. IEEE* **1995**, *83*, 378–406. [CrossRef]
10. Tan, J.C.M.; Cao, Q.; Quek, C. FE-RNN: A fuzzy embedded recurrent neural network for improving interpretability of underlying neural network. *Inf. Sci.* **2024**, *663*, 120276.
11. Zhou, K.; Oh, S.K.; Qiu, J.; Pedrycz, W.; Seo, K.; Yoon, J.H. Design of Hierarchical Neural Networks Using Deep LSTM and Self-organizing Dynamical Fuzzy-Neural Network Architecture. *IEEE Trans. on Fuzzy Syst.* **2024**, *32*, 2915–2929. [CrossRef]

12. Singh, B.; Doborjeh, M.; Doborjeh, Z.; Budhraja, S.; Tan, S.; Sumich, A.; Goh, W.; Lee, J.; Lai, E.; Kasabov, N. Constrained neuro fuzzy inference methodology for explainable personalised modelling with applications on gene expression data. *Sci. Rep.* **2023**, *13*, 456.
13. Lughofer, E.; Pratama, M. Evolving multi-user fuzzy classifier system with advanced explainability and interpretability aspects. *Inf. Fusion* **2023**, *91*, 458–476.
14. Pedrycz, W. Logic-based fuzzy neurocomputing with unineurons. *IEEE Trans. Fuzzy Syst.* **2006**, *14*, 860–873.
15. Souza, P.V.C. Regularized Fuzzy Neural Networks for Pattern Classification Problems. *Int. J. Appl. Eng. Res.* **2018**, *13*, 2985–2991.
16. Pedrycz, W.; Gomide, F. *An Introduction to Fuzzy Sets: Analysis and Design*; Mit Press: Cambridge, MA, USA, 1998.
17. Chicco, D.; Jurman, G. Survival prediction of patients with sepsis from age, sex, and septic episode number alone. *Sci. Rep.* **2020**, *10*, 17156.
18. Pedrycz, W. Neurocomputations in relational systems. *IEEE Trans. Pattern Anal. Mach. Intell.* **1991**, *13*, 289–297.
19. Klement, E.P.; Mesiar, R.; Pap, E. Triangular norms. Position paper III: Continuous t-norms. *Fuzzy Sets Syst.* **2004**, *145*, 439–454.
20. Klement, E.P.; Mesiar, R.; Pap, E. *Triangular Norms*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013; Volume 8.
21. Yager, R.R.; Rybalov, A. Uninorm aggregation operators. *Fuzzy Sets Syst.* **1996**, *80*, 111–120.
22. Lemos, A.; Caminhas, W.; Gomide, F. A fast learning algorithm for uninorm-based fuzzy neural networks. In Proceedings of the Fuzzy Information Processing Society (NAFIPS), 2012 Annual Meeting of the North American, Berkeley, CA, USA, 6–8 August 2012; pp. 1–6.
23. Klement, E.; Mesiar, R.; Pap, E. *Triangular Norms*; Kluwer Academic Publishers: Dordrecht, The Netherlands; Norwell, MA, USA; New York, NY, USA; London, UK, 2000.
24. Zhou, H.; Liu, X. Characterizations of (U2,N)-implications generated by 2-uninorms and fuzzy negations from the point of view of material implication. *Fuzzy Sets Syst.* **2020**, *378*, 79–102.
25. Lemos, A.; Caminhas, W.; Gomide, F. New uninorm-based neuron model and fuzzy neural networks. In Proceedings of the Fuzzy Information Processing Society (NAFIPS), 2010 Annual Meeting of the North American, Toronto, ON, Canada, 12–14 July 2010; pp. 1–6.
26. Wang, Y.; Lu, H.; Qin, X.; Guo, J. Residual Gabor convolutional network and FV-Mix exponential level data augmentation strategy for finger vein recognition. *Expert Syst. Appl.* **2023**, *223*, 119874. [CrossRef]
27. Yu, Y.F.; Zhong, G.; Zhou, Y.; Chen, L. FS-GAN: Fuzzy Self-guided structure retention generative adversarial network for medical image enhancement. *Inf. Sci.* **2023**, *642*, 119114. [CrossRef]
28. Li, L.J.; Zhou, S.L.; Chao, F.; Chang, X.; Yang, L.; Yu, X.; Shang, C.; Shen, Q. Model compression optimized neural network controller for nonlinear systems. *Knowl.-Based Syst.* **2023**, *265*, 110311. [CrossRef]
29. Khuat, T.T.; Gabrys, B. An online learning algorithm for a neuro-fuzzy classifier with mixed-attribute data. *Appl. Soft Comput.* **2023**, *137*, 110152. [CrossRef]
30. Hao, Y.; Yang, W.; Yin, K. Novel wind speed forecasting model based on a deep learning combined strategy in urban energy systems. *Expert Syst. Appl.* **2023**, *219*, 119636. [CrossRef]
31. Wang, L.; Li, H.; Hu, C.; Hu, J.; Wang, Q. Synchronization and settling-time estimation of fuzzy memristive neural networks with time-varying delays: Fixed-time and preassigned-time control. *Fuzzy Sets Syst.* **2023**, *470*, 108654. [CrossRef]
32. Su, Y.; Yang, C.; Qiao, J. Self-organizing pipelined recurrent wavelet neural network for time series prediction. *Expert Syst. Appl.* **2023**, *214*, 119215. [CrossRef]
33. Jan, N.; Gwak, J.; Pamucar, D.; MartÃnez, L. Hybrid integrated decision-making model for operating system based on complex intuitionistic fuzzy and soft information. *Inf. Sci.* **2023**, *651*, 119592. [CrossRef]
34. Hwang, C.L. Cooperation of robot manipulators with motion constraint by real-time RNN-based finite-time fault-tolerant control. *Neurocomputing* **2023**, *556*, 126694. [CrossRef]
35. Chen, Z.; Wu, K.; Wu, J.; Deng, C.; Wang, Y. Residual shrinkage transformer relation network for intelligent fault detection of industrial robot with zero-fault samples. *Knowl.-Based Syst.* **2023**, *268*, 110452. [CrossRef]
36. Chen, J.; Mao, C.; Song, W.W. QoS prediction for web services in cloud environments based on swarm intelligence search. *Knowl.-Based Syst.* **2023**, *259*, 110081. [CrossRef]
37. Dass, A.; Srivastava, S.; Kumar, R. A novel Lyapunov-stability-based recurrent-fuzzy system for the Identification and adaptive control of nonlinear systems. *Appl. Soft Comput.* **2023**, *137*, 110161. [CrossRef]
38. Chen, Y.; Wang, W.; Chen, X.M. Bibliometric methods in traffic flow prediction based on artificial intelligence. *Expert Syst. Appl.* **2023**, *228*, 120421. [CrossRef]
39. Hu, Z.; Cheng, Y.; Xiong, H.; Zhang, X. Assembly makespan estimation using features extracted by a topic model. *Knowl.-Based Syst.* **2023**, *276*, 110738. [CrossRef]
40. Gu, X. Self-adaptive fuzzy learning ensemble systems with dimensionality compression from data streams. *Inf. Sci.* **2023**, *634*, 382–399. [CrossRef]

41. Javaheri, D.; Gorgin, S.; Lee, J.A.; Masdari, M. Fuzzy logic-based DDoS attacks and network traffic anomaly detection methods: Classification, overview, and future perspectives. *Inf. Sci.* **2023**, *626*, 315–338. [CrossRef]
42. Kan, N.H.L.; Cao, Q.; Quek, C. Learning and processing framework using Fuzzy Deep Neural Network for trading and portfolio rebalancing. *Appl. Soft Comput.* **2024**, *152*, 111233. [CrossRef]
43. Singh, D.J.; Verma, N.K. Interval Type-3Â T-S fuzzy system for nonlinear aerodynamic modeling. *Appl. Soft Comput.* **2024**, *150*, 111097. [CrossRef]
44. Tavana, M.; Sorooshian, S. A systematic review of the soft computing methods shaping the future of the metaverse. *Appl. Soft Comput.* **2024**, *150*, 111098. [CrossRef]
45. Zhao, T.; Zhu, Y.; Xie, X. Topology structure optimization of evolutionary hierarchical fuzzy systems. *Expert Syst. Appl.* **2024**, *238*, 121857. [CrossRef]
46. Yan, A.; Wang, R.; Guo, J.; Tang, J. A knowledge transfer online stochastic configuration network-based prediction model for furnace temperature in a municipal solid waste incineration process. *Expert Syst. Appl.* **2024**, *243*, 122733. [CrossRef]
47. Barhaghtalab, M.H.; Sepestanaki, M.A.; Mobayen, S.; Jalilvand, A.; Fekih, A.; Meigoli, V. Design of an adaptive fuzzy-neural inference system-based control approach for robotic manipulators. *Appl. Soft Comput.* **2023**, *149*, 110970. [CrossRef]
48. Park, S.B.; Oh, S.K.; Kim, E.H.; Pedrycz, W. Rule-based fuzzy neural networks realized with the aid of linear function Prototype-driven fuzzy clustering and layer Reconstruction-based network design strategy. *Expert Syst. Appl.* **2023**, *219*, 119655. [CrossRef]
49. Pham, P.; Nguyen, L.T.; Nguyen, N.T.; Kozma, R.; Vo, B. A hierarchical fused fuzzy deep neural network with heterogeneous network embedding for recommendation. *Inf. Sci.* **2023**, *620*, 105–124. [CrossRef]
50. Cagcag Yolcu, O.; Yolcu, U. A novel intuitionistic fuzzy time series prediction model with cascaded structure for financial time series. *Expert Syst. Appl.* **2023**, *215*, 119336. [CrossRef]
51. Azizi, F.; Hamid, M.; Salimi, B.; Rabbani, M. An intelligent framework to assess and improve operating room performance considering ergonomics. *Expert Syst. Appl.* **2023**, *229*, 120559. [CrossRef]
52. Yu, D.; Fang, A.; Xu, Z. Topic research in fuzzy domain: Based on LDA topic modelling. *Inf. Sci.* **2023**, *648*, 119600. [CrossRef]
53. Zheng, K.; Zhang, Q.; Peng, L.; Zeng, S. Adaptive memetic differential evolution-back propagation-fuzzy neural network algorithm for robot control. *Inf. Sci.* **2023**, *637*, 118940. [CrossRef]
54. Stepin, I.; Suffian, M.; Catala, A.; Alonso-Moral, J.M. How to Build Self-Explaining Fuzzy Systems: From Interpretability to Explainability [AI-eXplained]. *IEEE Comput. Intell. Mag.* **2024**, *19*, 81–82. [CrossRef]
55. Alonso, J.M.; Castiello, C.; Mencar, C. Interpretability of fuzzy systems: Current research trends and prospects. In *Springer Handbook of Computational Intelligence*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 219–237.
56. de Campos Souza, P.V.; Lughofer, E.; Rodrigues Batista, H. An explainable evolving fuzzy neural network to predict the k barriers for intrusion detection using a wireless sensor network. *Sensors* **2022**, *22*, 5446. [CrossRef]
57. Jin, Y.; Cao, W.; Wu, M.; Yuan, Y.; Shi, Y. Simplification of ANFIS based on Importance-Confidence-Similarity Measures. *Fuzzy Sets Syst.* **2024**, *481*, 108887.
58. Pratama, M.; Lu, J.; Lughofer, E.; Zhang, G.; Er, M.J. An incremental learning of concept drifts using evolving type-2 recurrent fuzzy neural networks. *IEEE Trans. Fuzzy Syst.* **2016**, *25*, 1175–1192.
59. Li, Y.; Lin, Y.; Liu, J.; Weng, W.; Shi, Z.; Wu, S. Feature selection for multi-label learning based on kernelized fuzzy rough sets. *Neurocomputing* **2018**, *318*, 271–286.
60. de Campos Souza, P.V.; Lughofer, E. EFNN-NullUni: An evolving fuzzy neural network based on null-uninorm. *Fuzzy Sets Syst.* **2022**, *449*, 1–31.
61. Felizardo, L.K.; Lima Paiva, F.C.; de Vita Graves, C.; Matsumoto, E.Y.; Costa, A.H.R.; Del-Moral-Hernandez, E.; Brandimarte, P. Outperforming algorithmic trading reinforcement learning systems: A supervised approach to the cryptocurrency market. *Expert Syst. Appl.* **2022**, *202*, 117259. [CrossRef]
62. Dong, L.; Jiang, F.; Li, X.; Wu, M. IRI: An intelligent resistivity inversion framework based on fuzzy wavelet neural network. *Expert Syst. Appl.* **2022**, *202*, 117066. [CrossRef]
63. Hassani, S.; Dackermann, U.; Mousavi, M.; Li, J. A systematic review of data fusion techniques for optimized structural health monitoring. *Inf. Fusion* **2024**, *103*, 102136. [CrossRef]
64. Qin, B.; lai Chung, F.; Nojima, Y.; Ishibuchi, H.; Wang, S. Fuzzy rule dropout with dynamic compensation for wide learning algorithm of TSK fuzzy classifier. *Appl. Soft Comput.* **2022**, *127*, 109410. [CrossRef]
65. Liao, H.; He, Y.; Wu, X.; Wu, Z.; Bausys, R. Reimagining multi-criterion decision making by data-driven methods based on machine learning: A literature review. *Inf. Fusion* **2023**, *100*, 101970. [CrossRef]
66. de Campos Souza, P.V.; Lughofer, E. EFNC-Exp: An evolving fuzzy neural classifier integrating expert rules and uncertainty. *Fuzzy Sets Syst.* **2023**, *466*, 108438. [CrossRef]
67. Mirjalili, S.; Mirjalili, S.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
68. Pizzileo, B.; Li, K.; Irwin, G.W.; Zhao, W. Improved Structure Optimization for Fuzzy-Neural Networks. *IEEE Trans. Fuzzy Syst.* **2012**, *20*, 1076–1089. [CrossRef]

69. Gómez-Skarmeta, A.; Jiménez, F.; Sánchez, G. Improving interpretability in approximative fuzzy models via multiobjective evolutionary algorithms. *Int. J. Intell. Syst.* **2007**, *22*, 943–969. [CrossRef]
70. Zhao, W.; Li, K.; Irwin, G. A New Gradient Descent Approach for Local Learning of Fuzzy Neural Models. *IEEE Trans. Fuzzy Syst.* **2013**, *21*, 30–44. [CrossRef]
71. Oh, S.K.; Pedrycz, W.; Park, H.S. Genetically Optimized Fuzzy Polynomial Neural Networks. *IEEE Trans. Fuzzy Syst.* **2006**, *14*, 125–144. [CrossRef]
72. Cao, B.; Zhao, J.; Lv, Z.; Gu, Y.; Yang, P.; Halgamuge, S. Multiobjective Evolution of Fuzzy Rough Neural Network via Distributed Parallelism for Stock Prediction. *IEEE Trans. Fuzzy Syst.* **2020**, *28*, 939–952. [CrossRef]
73. Ebadzadeh, M.; Salimi-Badr, A. IC-FNN: A Novel Fuzzy Neural Network with Interpretable, Intuitive, and Correlated-Contours Fuzzy Rules for Function Approximation. *IEEE Trans. Fuzzy Syst.* **2018**, *26*, 1288–1302. [CrossRef]
74. Heydari, A.; Nezhad, M.M.; Neshat, M.; Garcia, D.A.; Keynia, F.; Santoli, L.D.; Tjernberg, L.B. A Combined Fuzzy GMDH Neural Network and Grey Wolf Optimization Application for Wind Turbine Power Production Forecasting Considering SCADA Data. *Energies* **2021**, *14*, 3459. [CrossRef]
75. Le, T.L.; Huynh, T.T.; Hong, S.H. A Modified Grey Wolf Optimizer for Optimum Parameters of Multilayer Type-2 Asymmetric Fuzzy Controller. *IEEE Access* **2020**, *8*, 121611–121629. [CrossRef]
76. Qin, H.; Meng, T.; Cao, Y. Fuzzy Strategy Grey Wolf Optimizer for Complex Multimodal Optimization Problems. *Sensors* **2022**, *22*, 6420. [CrossRef]
77. Sánchez, D.; Melin, P.; Castillo, O. A Grey Wolf Optimizer for Modular Granular Neural Networks for Human Recognition. *Comput. Intell. Neurosci.* **2017**, *2017*, 4180510. [CrossRef]
78. de Campos Souza, P.V.; Silva, G.R.L.; Torres, L.C.B. Uninorm based regularized fuzzy neural networks. In Proceedings of the 2018 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS), Rhodes, Greece, 25–27 May 2018; pp. 1–8.
79. de Campos Souza, P.V.; de Oliveira, P.F.A. Regularized fuzzy neural networks based on nullneurons for problems of classification of patterns. In Proceedings of the 2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), Penang, Malaysia, 28–29 April 2018; pp. 25–30.
80. Miller, G.A. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychol. Rev.* **1956**, *63*, 81. [CrossRef]
81. Unwin, A.; Kleinman, K. The Iris Data Set: In Search of the Source of Virginica. *Significance* **2021**, *18*, 26–29. . [CrossRef]
82. Elter, M. Mammographic Mass. UCI Machine Learning Repository, 2007. Available online: https://archive.ics.uci.edu/dataset/161/mammographic+mass (accessed on 27 March 2025).
83. Haberman, S. Haberman's Survival. UCI Machine Learning Repository, 1999. Available online: https://archive.ics.uci.edu/dataset/43/haberman+s+survival (accessed on 27 March 2025).
84. Yeh, I.C. Blood Transfusion Service Center. UCI Machine Learning Repository, 2008. Available online: https://archive.ics.uci.edu/dataset/176/blood+transfusion+service+center (accessed on 27 March 2025).
85. Liver Disorders. UCI Machine Learning Repository, 1990. Available online: https://archive.ics.uci.edu/dataset/60/liver+disorders (accessed on 27 March 2025).
86. Khozeimeh, F.; Jabbari Azad, F.; Mahboubi Oskouei, Y.; Jafari, M.; Tehranian, S.; Alizadehsani, R.; Layegh, P. Intralesional immunotherapy compared to cryotherapy in the treatment of warts. *Int. J. Dermatol.* **2017**, *56*, 474–478. [CrossRef] [PubMed]
87. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
88. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
89. Chan, T.F.; Golub, G.H.; LeVeque, R.J. Algorithms for computing the sample variance: Analysis and recommendations. *Am. Stat.* **1983**, *37*, 242–247. [CrossRef]
90. St, L.; Wold, S. Analysis of variance (ANOVA). *Chemom. Intell. Lab. Syst.* **1989**, *6*, 259–272.
91. Abdi, H.; Williams, L.J. Tukey's honestly significant difference (HSD) test. *Encycl. Res. Des.* **2010**, *3*, 1–5.
92. Razali, N.M.; Wah, Y.B. Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests. *J. Stat. Model. Anal.* **2011**, *2*, 21–33.
93. Gastwirth, J.L.; Gel, Y.R.; Miao, W. The Impact of Levene's Test of Equality of Variances on Statistical Theory and Practice. *Stat. Sci.* **2009**, *24*, 343–360. [CrossRef]
94. Kaya, U.; Yılmaz, A.; Aşar, S. Sepsis Prediction by Using a Hybrid Metaheuristic Algorithm: A Novel Approach for Optimizing Deep Neural Networks. *Diagnostics* **2023**, *13*, 2023. [CrossRef] [PubMed]
95. Goh, K.H.; Wang, L.; Yeow, A.Y.K.; Poh, H.; Li, K.; Yeow, J.J.L.; Tan, G.Y.H. Artificial intelligence in sepsis early prediction and diagnosis using unstructured data in healthcare. *Nat. Commun.* **2021**, *12*, 711. [PubMed]
96. Jones, L.C.; Dion, C.; Efron, P.A.; Price, C.C. Sepsis and cognitive assessment. *J. Clin. Med.* **2021**, *10*, 4269. [CrossRef]
97. Xie, Y.; Li, B.; Li, Y.; Shi, F.; Chen, W.; Wu, W.; Zhang, W.; Fei, Y.; Zou, S.; Yao, C. Combining blood-based biomarkers to predict mortality of sepsis at arrival at the Emergency Department. *Med Sci. Monit. Int. Med J. Exp. Clin. Res.* **2021**, *7*, e929527-1–e929527-8.

98. Gamboa-Antiñolo, F.M. Prognostic tools for elderly patients with sepsis: In search of new predictive models. *Intern. Emerg. Med.* **2021**, *16*, 1027–1030.
99. Fonseca, F.S.; Torcate, A.S.; Silva, A.C.G.D.; Freire, V.H.W.; Farias, G.P.D.M.D.; Oliveira, J.F.L.D. Early prediction of generalized infection in intensive care units from clinical data: A committee-based machine learning approach. In Proceedings of the 2022 IEEE Latin American Conference on Computational Intelligence (LA-CCI), Montevideo, Uruguay, 23–25 November 2022; pp. 1–6.
100. Samuel, S.V.; Viggeswarpu, S.; Chacko, B.; Belavendra, A. Predictors of outcome in older adults admitted with sepsis in a tertiary care center. *J. Indian Acad. Geriatr.* **2023**, *19*, 105–113.
101. Shanthi, N.; A, A. A novel machine learning approach to predict sepsis at an early stage. In Proceedings of the 2022 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 25–27 January 2022; pp. 1–7.
102. John, G.H.; Langley, P. Estimating continuous distributions in Bayesian classifiers. In Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, 18–20 August 1995; pp. 338–345.
103. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297.
104. Walker, S.H.; Duncan, D.B. Estimation of the probability of an event as a function of several independent variables. *Biometrika* **1967**, *54*, 167–179. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.